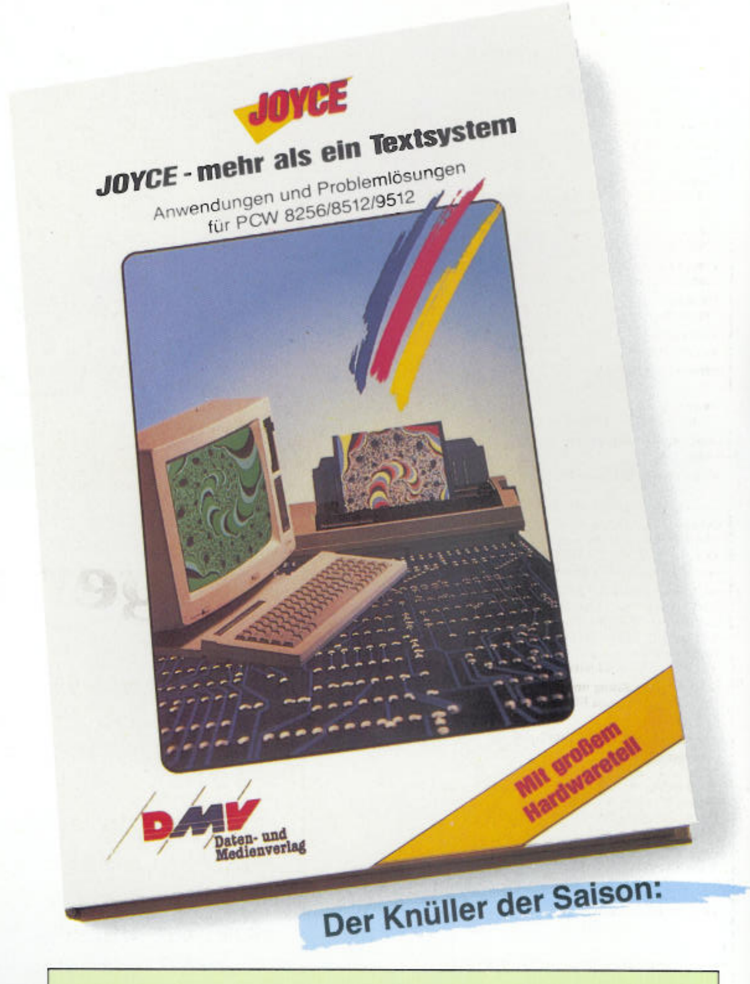


6,- DM öS 50,- sfr 6,-
CPC
Amstrad
INTERNATIONAL
CPC · PCW JOYCE · PC
2
Februar 1990
6.Jahrgang
PC:
- Tolle Programme zum Abtippen
- Einführung in MS-DOS
- Tips zum NVR
CPC-Hits:
- neue Schriften für LQ3500
- Rechnungen schreiben
- logisches Denkspiel
- neue Pokes und Calls
- Tips zu Multiplan
So geht's:
Alles über Kopierschutz
PCW:
- Terminkalender mit Pfiff
- Aus Assembler wird BASIC
- Juca-Adreßverwaltung komplett
Viele Tips & Tricks für alle Amstrad-Computer
DISCOLOGY

Das Buch zum JOYCE!

Alle Besitzer eines PCW 8256/8512/9512, die ahnten, daß der Horizont ihres Computers weit über LocoScript hinausgeht, finden jetzt die Bestätigung:

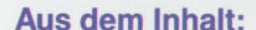


Der Knüller der Saison:

JOYCE
– mehr als ein Textsystem

Die Autoren, in der JOYCE-Szene keine Unbekannten, haben auf über 300 Seiten alles Wissenswerte über die "andere Seite" der PCWs zusammengetragen. Anfänger wie Profis, Anwender wie Programmierer finden in diesem Buch einen reichen Schatz an Tips und Tricks sowie ein unentbehrliches Nachschlagewerk. Erstmals wird auch in einem Buch zum PCW ein "heißes Eisen" ausführlich behandelt: die Hardware. Dieses Kapitel zeigt völlig neue Möglichkeiten des JOYCE – so wird beispielsweise der Anschluß eines Sprachsynthesizers und der Selbstbau einer Schnittstelle besprochen. Ausführliche Kapitel über BASIC und LOGO erlauben Ihnen, diese Sprachen auszureizen: Anfänger, die sich erstmals in der Programmierung versuchen wollen, finden durch sinnvolle kurze Beispielprogramme wertvolle Unterstützung. Ein besonderes "Schmankerl" ist das große Extra-Kapitel zur JETSAM-Dateiverwaltung!

Platinenlayouts im Buch **69,– DM***



Sprachen:
LOGO als Grafiker
komplette Befehlsübersicht
BASIC: Erläuterungen aller Befehle mit Beispielprogrammen
JETSAM: Generator für JETSAM-Verarbeitung!
Kurzübersicht: Turbo PASCAL & C

Programmierhilfen, Tips und Tricks:
Steuercode-Tabellen für Bildschirm und Drucker
XBIOS-Routinen
OUTs und POKEs unter BASIC

Hardware
Speichererweiterung
Zweitlaufwerk
Druckkopfreinigung
Bildschirminverter
Schnittstelle am Expansionsport
Sprachsynthesizer

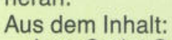
324 Seiten mit farbigen Abbildungen
Hardcover, gebunden
ISBN 3-926177-02-0

Weiterhin erhältlich:
doppelte Platine, geprüft
3-Zoll-Disketten mit allen Dateien und Programmen
Bestellkarte im Buch

Ebenfalls von DMV

Praktische Textverarbeitung mit Joyce

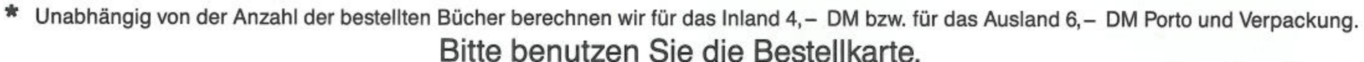
Ein Buch/Disketten-Paket. Der Autor Jürgen Siebert zeigt in diesem Buch Möglichkeiten der Textverarbeitung auf, die Sie von LocoScript nicht erwartet hätten... Von der Pike auf werden Sie an den Umgang mit Schablonen und Standardlayout herangeführt. Einige Abstecher führen Sie anhand anschaulicher Beispiele an Textverarbeitung und CP/M (ED/Wordstar) heran.



- LocoScript Spezial - Softwaretraining für Fortgeschrittene
- Fehler im System: Wie rette ich meinen Text?
- Joyce-Tasteninstallationsdatei für das Programm Wordstar
- Aleatorische Poetik: Der Computer dichtet

Auf Diskette: über 50 Dateien mit Schablonen, Briefen, Postkarten, Serien-Rundschreiben, Formularen, Etiketten, Druckbeispielen, Schriften, Bildschirm-Installationen und vieles mehr.

Leinen-Hardcover, 207 Seiten, 3-Zoll-Diskette Best.-Nr. 401 **DM 89,–***



Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege



Impressum

Herausgeber
Christian Widuch

Chefredakteur
Stefan Ritter

Stv. Chefredakteur
Michael Ebbrecht (me)

Redaktion
Claus Daschner (cd), Joachim Freiburg (jf)

Redaktions-Assistenz
Anke Kerstan (ke), Susanne Eska (es)

Schlußredaktion
Vera Brinkmann

Produktionsleitung
Gerd Köberich

Bereichsleitung
Claudia Ebbrecht (Fotosatz/Lektorat)
Margarete Schenk, Helmut Skoupy (Montage/Reprografie)

Layout
Michael Grebenstein

Fotografie
Christian Heckmann, Heike Reifer

Fotosatz
Marcus Geppert

Lektorat
Susanne Lessinger, Angela Palm

Montage/Reprografie
Dieter Schnobl, Andrea Gundlach

Werbegestaltung
Mohamed Hawa, Petra Küch

Anzeigenverkaufsleitung
Wolfgang Schnell

Anzeigenverkauf für PLZ 1, 4, 5
Gerlinde Rachow, Telefon: (0 56 51) 80 93 90
Wolfgang Brill, Telefon: (0 56 51) 80 93 80

Anzeigenverkauf für PLZ 2 + 3
DMV-Verlagsbüro Hamburg
Ohlsdorfer Straße 34, 2000 Hamburg 60
Sylvia Ehrenpfordt, Telefon: (0 40) 46 12 33, Telefax: (0 40) 47 43 10

Anzeigenverkauf für PLZ 6 – 8
DMV-Verlagsbüro München
Zaunkönigweg 2c, 8000 München 82
Telefon: (0 89) 4 39 10 87, Telefax: (0 89) 4 39 10 80
Leitung: Britta Fiebig
Anzeigenverkauf: Monika Schöbel, Jens Dhein, Peter Schätzle

Anzeigenverwaltung und Disposition
Andrea Giese, Karina Ehrlich, Beate Kranz

Anzeigenpreise
Es gilt die Anzeigenpreisliste Nr. 4 vom 01. 01. 1989.

Anzeigengrundpreise
1/1 Seite sw DM 5240,–
Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 750,–.
Vierfarbzuschlag DM 2250,–.

Anschrift Verlag/Redaktion:
DMV Daten und Medien Verlag
Widuch GmbH & Co. KG
Fuldaer Straße 6
3440 Eschwege
Telefon: (0 56 51) 809–0
Telefax: (0 56 51) 80 93 33

Vertrieb
Verlagsunion Erich Pabel – Arthur Moewig KG (VPM)
Friedrich-Bergius-Straße 20
6200 Wiesbaden

Druck
Druckerei Jungfer, 3420 Herzberg

Bezugspreise
»PC Amstrad International« erscheint monatlich am Ende des Vormonats.
Einzelpreis DM 6,–/sfr. 6,–/öS 50,–

Abonnementpreise
Die Preise verstehen sich grundsätzlich einschließlich Porto und Verpackung.

Inland:
12 Ausgaben: DM 66,–
6 Ausgaben: DM 33,–

Europäisches Ausland:
12 Ausgaben: DM 96,–
6 Ausgaben: DM 48,–

Außereuropäisches Ausland:
12 Ausgaben: DM 120,–
6 Ausgaben: DM 60,–

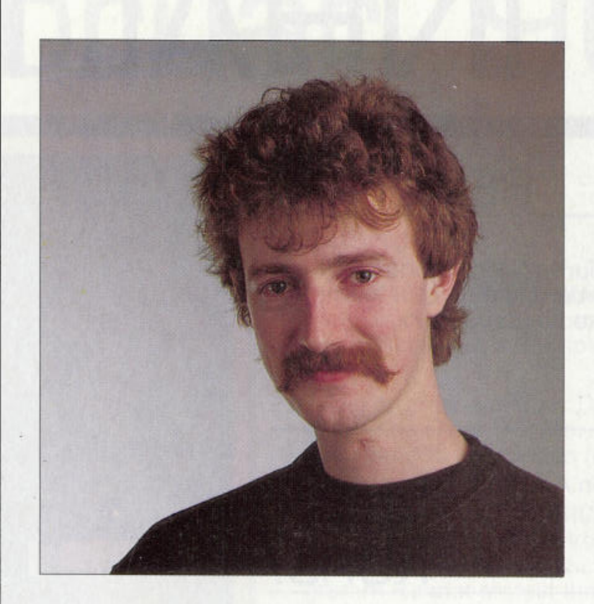
Bankverbindungen:
Postscheck Frankfurt/M: Kto.-Nr.: 230 43–608
Raiffeisenbank Eschwege:
BLZ: 522 603 85, Kto.-Nr.: 245 7008

Die Abonnementbestellung kann innerhalb einer Woche nach Auftrag beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich widerrufen werden. Zur Wahrung der Frist reicht der Poststempel. Das Abonnement verlängert sich automatisch um 6 bzw. 12 Ausgaben, wenn es nicht mindestens 6 Wochen vor Ablauf beim Verlag schriftlich gekündigt wird.

Die Gesamtauflage enthält einen Beihefter des Interest-Verlages.

Mitglied der Informationsgemeinschaft zur Feststellung der Verbreitung von Werbeträgern e.V. (IVW), Bad Godesberg.
ISSN 0935-9095





Jäger und Sammler

Ein nicht zu leugnender Urtrieb in uns allen ist der stete Drang zu jagen und zu sammeln. Da die Jägerei in der heutigen Zeit durch die geballte Häufung allesbietender Einkaufszentren in den Hintergrund verdrängt wurde, mußte die brodelnde Urkraft ihre Prioritäten zwangsläufig auf das Sammeln verlegen.

Die Erfindung des Computers und der damit unendlich großen Menge reproduzierbarer Software führte folglich dazu, daß sich gewisse Kneipensprüche ("Ich habe da eine tolle Briefmarkensammlung zu Hause.") im Inhalt veränderten ("Darf ich Dir vielleicht meine Softwaresammlung zeigen?").

Da das Sammeln von Software im allgemeinen ein sehr kostspieliges Unterfangen ist, wird kopiert, solange der Diskettenplatz ausreicht. Den Herstellern der Programme kann das selbstverständlich nicht recht sein, so daß sie nach Möglichkeiten sinnen, diese Aktionen zu unterbinden. Erstaunlich ist, daß bei einer so großen Anzahl kluger Köpfe ausgerechnet die teuerste und unpraktischste Art der Gegenmaßnahmen am beliebtesten zu sein scheint: der softwaremäßige Kopierschutz. Für legale Besitzer ist dieser Schutz eine unnötige Schikane, für illegale Besitzer nur kurze Zeit ein ernsthaftes Hindernis an der Nutzung der Programme.

Letzteres ist der Grund dafür, daß der Kopierschutz immer komplizierter, teurer und anfälliger für Fehler wird. Da die "Hacker" aber auch nicht dümmer werden, droht ein unsinniger Kleinkrieg zwischen "Schützern" und "Hackern" zu eskalieren, bei dem natürlich wieder einmal der Normalverbraucher in die berühmte Röhre schaut.

Wir wollen uns in dieser Ausgabe näher mit den Möglichkeiten des Kopierschutzes befassen und hoffen, daß sich das Motto "Der beste Kopierschutz ist ein 300 Seiten starkes Handbuch" auch bei den Machern von Computersoftware durchsetzen wird.

Ihr

Joachim Freiburg

Joachim Freiburg

INHALT

BERICHTE:

Kopierschutzmechanismen 8
– Der Kopierschutz: Wie funktioniert er? Was bringt er? Kann man ihn rechtfertigen? Lesen Sie die Antworten in unserem Bericht.

SERVICE:

31 **AMS-Line**
– Neuigkeiten von Amstrad

KURS:

dBase verständlich 26
– Grundlagen zum Datenbanksystem (Teil 5)

PROGRAMME:

32 **Verwaltungshilfe**
– Rechnungen schreiben leichtgemacht. Lassen Sie Ihren CPC für sich arbeiten.

41 **Sly Sprite**
– Strategie ist Trumpf bei dieser Zahlenhatz.

ASSEMBLER:

Die Assembler-Ecke 44
– Optimierung von Assemblerprogrammen

TIPS & TRICKS:

48 **100,– DM für 1 kByte**
– CPC-Programme, short and sweet

52 **Kleine Ursache, große Wirkung**
– Neue Poke- und Call-Befehle für den CPC

54 **Ein Drucker lernt schreiben**
– Download Fonts für LQ3500

58 **Rolle vor- und rückwärts**
– Bildaufbau vom Feinsten

59 **Aus zwei mach eins**
– Wertvolle Tips zu Multiplan

SOFTWARE REVIEWS:

Epyx Action 14
– Fünf auf einen Streich in einer neuen Compilation

Fighting Soccer 17
– Fußball ... und kein Ende

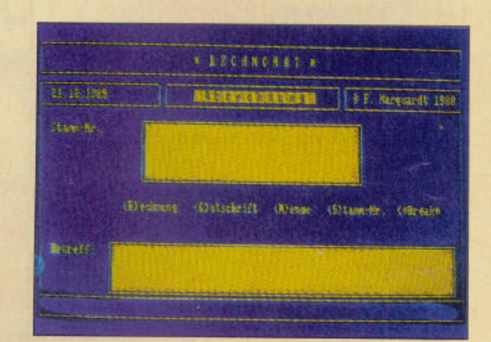
Assembler von der Pike auf 18
– Der Assemblerkurs von Sybex

ABENTEUER:

20 **Gamers Message**
– Tips zu kniffligen Spielen

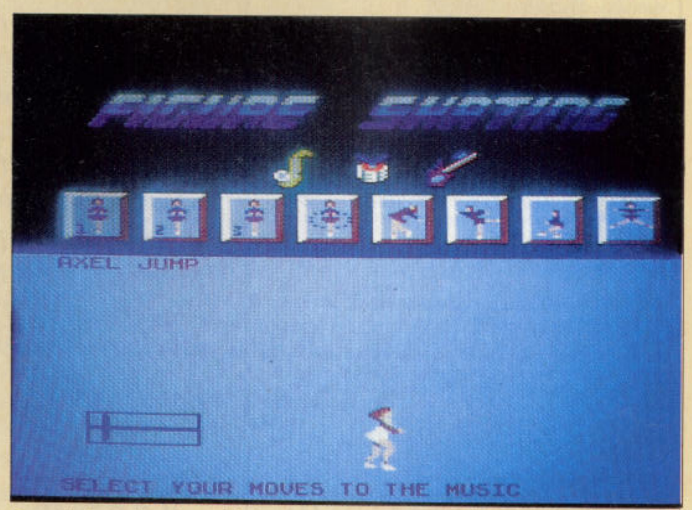
Ein Dorn im Auge legaler Anwender: der Kopierschutz

S. 8



Das Bezahlen Ihrer Rechnungen nimmt Ihnen der CPC noch nicht ab, das Schreiben allerdings schon

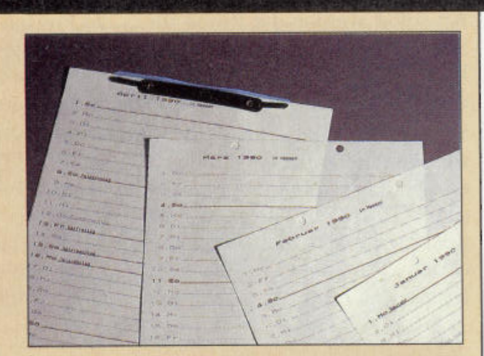
S. 32



Epyx bringt eine Compilation von starken Actionspielen. Lesen Sie dazu unseren Test

S. 14

Februar '90



Mit unserem Kalenderprogramm haben Sie Ihre Termine Monat für Monat im Griff

S. 72



Sierras Neue: Die Stärken und Schwächen von Hero's Quest und Larry III wollen wir Ihnen nicht vorenthalten

S. 88



Mit unserem Listing Upcase sind Sie in der Lage, Ihre Programme auf einfache Weise leserlicher zu gestalten

S. 82

PCW:

JUCA Teil 3 60
– Im dritten und letzten Teil der Serie erhalten Sie alle benötigten Informationen zu dem Druckmenü sowie zu der Schnittstelle von dBase II zu LocoScript 2 und LocoMail 2.

VDE 63
– In der englischen Public Domain gibt es den Video-Display-Editor, welcher sich hervorragend zum Arbeiten mit dem PCW eignet.

Von Assembler nach BASIC 66
– Können Sie als BASIC-Programmierer auf Maschinenroutinen verzichten? Das Programm Hexdat erstellt Ihnen Dataloder für BASIC.

Zeitgeschichte 68
– Wollen Sie immer genau wissen, was die Uhr geschlagen hat? Benutzen Sie unser Utility UHR.BAS, welches Sie ohne Probleme in eigene Programme einbinden können.

Alle Termine im Blick 72
– Wenn Sie bis jetzt noch keinen Kalender und Terminplaner für das neue Jahr besitzen, wird es langsam Zeit. Drucken Sie sich diesen doch selbst aus.

PC:

76 **Datenmillionär**
– Festplatte oder Filecard?

78 **SaveDir**
– Test des Schutzprogramms

80 **Use Your Mouse**
– Stärken und Schwächen des Maus-Utilities

81 **Wider die Datenmanipulation**
– Dem Datenmanipulierer ein Schnippchen geschlagen! Kleiner Eingriff - große Wirkung.

82 **Strukturierter Programmcode**
– Bringen Sie Form in Ihre Programme. "Upcase" erledigt das für Sie.

88 **The Quest goes on**
– Die "Neuen" von Sierra

90 **NVR durchleuchtet (Teil 1)**
– Der Aufbau im Detail

93 **MS-DOS, die ersten Berührungen**
– AUTOEXEC.BAT und Stapelverarbeitung

96 **JR in die Karten geschaut**
– Präsident werden, das will jeder. Mit "For President..." können Sie Ihre Wünsche wahr werden lassen.

RUBRIKEN:

Editorial 3
Impressum 3
Aktuell 6
Leserbriefe 22
Händlerverzeichnis 99
Kleinanzeigen 101
Inserentenverzeichnis 102
Vorschau 102

Lernprogramme für PCW



Aus dem Hause Düsi Software kommen vier Programme, mittels derer Sie jedes Lerngebiet vollkommen abdecken können.
Der Vokabeltrainer beinhaltet alle von Vokabeltrainern bekannten Funktionen wie Laden/Speichern, Eingeben, Lernen und Ansehen. Gelernt werden kann grundsätzlich jede Sprache, die Ihr Joyce darstellen kann.
Der Verbentrainer ist speziell auf die Verben der englischen Sprache zugeschnitten. Hier erfolgte eine sinnvolle Unterteilung des Lernstoffes in drei verschiedene Gebiete.
Mit dem Auswahltrainer steht Ihnen nun eine etwas andere Art des Lernens zur Verfügung. Hier werden Ihnen – nachdem Sie die Fragen und Antworten eingegeben haben – für jede Frage vier Antworten angegeben, von denen Sie sich die richtige heraussuchen müssen.
Der Lerntrainer stellt in der Düsi-Reihe das flexibelste Programm dar. Hier können selber sogenannte Masken definiert werden, wo Sie genau festlegen können, wie große und wie viele Felder Sie haben möchten.
Preis: zwischen 49 und 79 DM
Info:
Düsi Software,
FA. Daniel Schwinn,
Meisenweg 6,
7073 Lorch

Neue Art der Serviceleistung

Eine bisher eher unbekannte Art der Serviceleistung bietet die Firma Alpha 2000 jetzt an.

Haben Sie sich für einen Computerzusatz entschieden, brauchen Sie jetzt nicht mehr unendlich viele Anzeigen in Zeitschriften durchzuschauen, um da billigste Angebot herauszufinden. Für 18,– DM können Sie bei Alpha 2000 den momentan billigsten Händler herausfinden. Als Schutz für Sie bietet Alpha 2000 eine Geld-zurück-Garantie, die in Anspruch genommen werden kann, sobald Sie einen Nachweis für ein billigeres Angebot erbracht haben.
Info:
Alpha 2000 GmbH,
Ingolstädter Str. 27,
6000 Frankfurt/M

Neue Funktionen bei BATCOM

Bei BATCOM handelt es sich um einen Compiler, der Batch-Dateien in EXE-Dateien compiliert.
Die neue Version 2.2 enthält jetzt Funktionen für Schleifenbildungen, Übergaben von Parametern an Unterprogramme sowie für zeilenweises Einlesen von Dateien.
Des weiteren verfügt das Programm über eine Schnittstelle zu Programmroutinen, die in C oder Assembler erzeugt wurden.
Preis: 188,– DM
Info:
Software-Vertrieb Brieskorn,
Am Rüppurrer Schloß 5/267,
7500 Karlsruhe 51

Ashton-Tate senkt Preise für Schulversionen

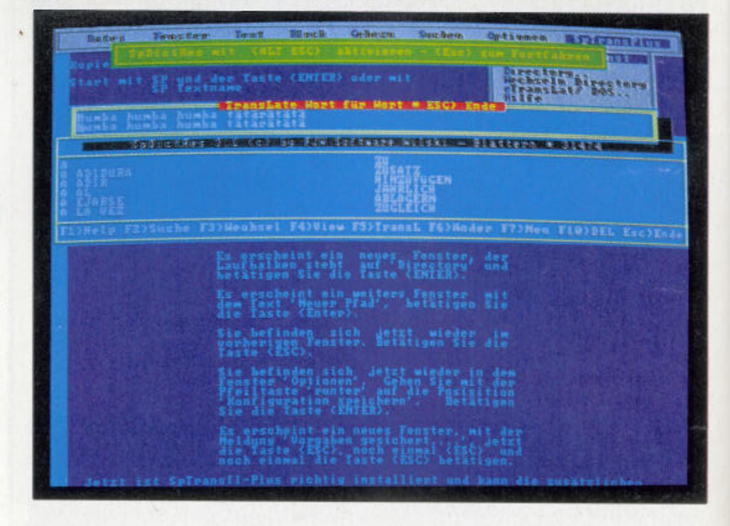
Ab sofort werden die Preise für Schulversionen und Bildungspakete aus dem Hause Ashton-Tate drastisch gesenkt.

Schüler, Studenten sowie nahezu alle anerkannten Bildungseinrichtungen können die gesamte Softwarepalette, die Programme wie beispielsweise dBase, Framework sowie Full Impact und FullPaint beinhaltet, zu Preisen beziehen, die 70 Prozent unter dem empfohlenen Verkaufspreis liegen. Ashton-Tate weist darauf hin, daß es sich bei diesen Programmen um keine eingeschränkten Vollprodukte handelt.
Info:
Ashton Tate,
Hahnstraße 70
6000 Frankfurt/M.

'Faxen machen' mit Ihrem PC

Von AMSTRAD gibt es wieder einmal eine Neuigkeit. Wer das ewige Anwählen, Warten und Bestätigen beim Faxen nicht mehr auf sich nehmen will, kann hierzu bald seinen PC in Verbindung mit dem FX 9600 T benutzen.
Das Personal Telefax FX 9600 T kann mehrere Aufgaben in Ihrem Büro wahrnehmen. Es kann entweder als Telefax, Telefon, Kopierer oder aber als Thermodrucker eingesetzt werden und erspart somit eine Menge Platz in Ihrem Büro. In der Funktion als Thermodrucker stehen drei Emulationen als Epson FX, Proprinter und IBM Quit zur Verfügung.
Info:
AMSTRAD GmbH,
Dreieichstraße 8,
6082 Mörfelden

Computer weltweit



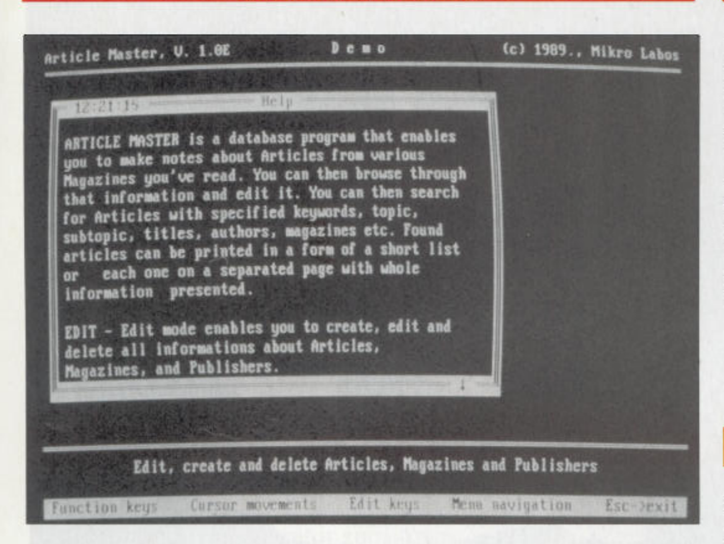
Spdict heißt das Programm, das Sie entweder als Wörterbuch oder als Lexikonprogramm einsetzen können. Hierbei braucht das Lexikonprogramm nicht nur für Fremdsprachen eingesetzt werden. Es ist auch möglich, es spezieller einzusetzen (Abkürzungen, Ärztelexikon).
Pro Lexikon können über 4 Milliarden Stichpunkte bei einer maximalen Länge von 253 Zeichen pro Stichpunkt verwaltet werden. Der Lieferumfang des Programms Spdict enthält einen ASCII-Compiler und einen speicherresidenten Teil mit einem englischen Grundwortschatz (ca. 100000 Wörter).

Nachträglich können Sie zwei weitere Wortschatz-Lexika für Französisch-Deutsch und Spanisch-Deutsch erwerben.

Preis: 98,– DM

Info:
FJW Wonderworks
Software-Entwicklung,
Zwickauer Damm 12,
1000 Berlin 47

Artikelarchivierung



Hinter dem Namen Article Master versteckt sich ein einfach zu bedienendes Programm, welches auf die Verwaltung von Artikeln oder spezieller Fachlektüre zugeschnitten ist. In dem Programm können die Angaben 'Artikel', 'Magazin' sowie 'Verlag' gemacht werden. Besonders zeichnet sich das Programm jedoch durch seine gut gewählte Bildschirmaufteilung sowie durch eine gute Suchfunktion an.

Preis: 85,– DM

Info:
K+S
Computer-Peripheriegeräte,
Siebenbrunnen 21,
A-1050 Wien

PizazzPlus 1.3 jetzt erhältlich

Die neue Version von Pizazz, einem universellen Hardcopy-Programm, das eine Erweiterung zu der PrtSc-Taste darstellt, wurde jetzt erweitert.

Insgesamt werden jetzt über 300 Drucker unterstützt. Neben Schwarzweißausdrucken sind auch Farbausdrucke möglich, wobei die Anordnung der Farben frei gewählt werden kann.

Ein ausführlicher Test wird in der nächsten Ausgabe der PC-Amstrad erscheinen.

Preis: 339,– DM

Info:
Peter Rosental GmbH,
Gabelsbergstraße 36,
8000 München 2

GCPM

Die bekannte Benutzeroberfläche GCPM für den PCW gibt es jetzt in einer verbesserten Version.

Enthalten sind Funktionen zum Kopieren, Löschen und Umbenennen von Dateien sowie zum Verschlüsseln und Entschlüsseln von ASCII-Texten. Neben dem Programm mit einfach zu bedienender Installationsroutine wird ferner ein ausführliches Handbuch mitgeliefert. GCPM unterstützt neben der Tastatur auch die weit verbreitete Gerdes-Maus. Dies muß in dem Installationsprogramm angegeben werden.

Preis: 69,– DM

Info:
DMV-Software,
Fuldaer Str. 6,
3440 Eschwege

Kommunikation

Von der Firma Transmodul aus Saarbrücken gibt es interessante Neuigkeiten. Das Modemprogramm T&T, welches für einen Preis von 99,– DM erhältlich ist, unterstützt AT sowie CCITT-Modems.

Durch Anwählen bestimmter Menüpunkte lassen sich Funktionen wie zeitverzögertes Senden, Wahlwiederholung und Datei senden/empfangen aufrufen. Interessant ist die integrierte Adreßdatenbank. Sind die Daten einmal eingegeben, können Sie, ohne manuelle Veränderungen an Übertragungsgeschwindigkeit oder -modus machen zu müssen, gleich durch Anwahl des bestimmten Partners die Verbindung aufnehmen. Ist die Sitzung gelaufen, können Sie sich das Sitzungsprotokoll noch einmal in aller Ruhe anschauen.

Der Host-Modus versetzt Sie in die Lage, von jeder beliebigen Stelle - wo immer Sie sich auch befinden - auf Ihre Daten, welche zu Hause im PC auf Sie warten, zuzugreifen.

Selbstverständlich können dies auch andere tun. Damit nicht jeder diese Möglichkeit hat, können Sie dies natürlich über ein Paßwort einschränken. Eine weitere Möglichkeit von T&T ist es, Daten zu beliebigen vorprogrammierten Zeiten an einen Empfänger abzusenden, ohne daß Sie anwesend sein müssen.

Info:
Transmodul,
Am Staden 18,
6600 Saarbrücken

TextMaker 2.01



Die Textverarbeitung TextMaker wird in einer erweiterten Version angeboten.

Hinzugekommen ist jetzt, die Netzwerkfähigkeit. TextMaker kann nun unter allen Novell-Netzwerken eingesetzt werden. Zu diesem Zeitpunkt sind mehr als 550 Druckertreiber für TextMaker erhältlich. So ist es doch recht wahrscheinlich, daß auch Ihr Drucker von TextMaker unterstützt wird.

Preis: 248,– DM

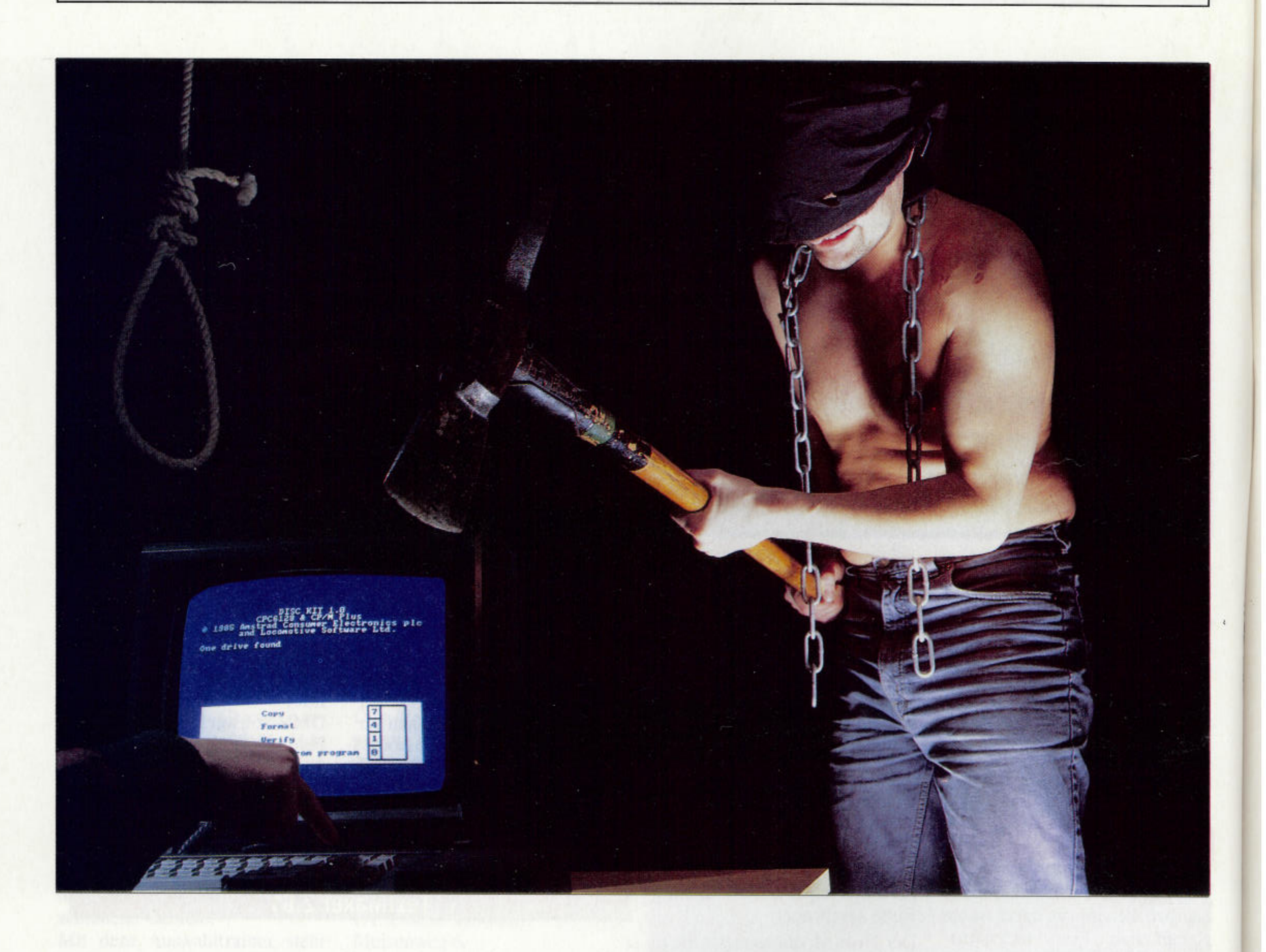
Info:
SoftMaker Software GmbH,
Grabbestraße 9,
8500 Nürnberg 90

Flugsimulator in neuer Version

Der bekannte Flugsimulator von Microsoft liegt jetzt in der Version 4.0 vor. Bei dieser Version wurde das Erstellen eigener Flugzeuge integriert.

Das komplexe Programm verfügt nun über 118 Flughäfen. Desweiteren sind im Lieferumfang die Flugzeuge Cessna 182, Gates Business Jet, Sopwith Camel sowie Schweizer 2-32 enthalten.

Weitere Informationen erfragen Sie bitte bei Ihrem Fachhändler.



Kopierschutzmechanismen

Aufbau und Anwendung

Sicherheitskopien (sogenannte Backups) sind in modernen Rechenzentren an der Tagesordnung. Es wäre nicht auszudenken, wenn wichtige Daten durch einen Programmfehler oder – auch nicht auszuschließen – durch einen Plattencrash irgendwo jenseits von Gut und Böse landen würden... Ist es doch passiert, kann man im Notfall immer noch auf die letzte Sicherung zurückgreifen.

Jeder vernünftige Computer-Hobbyist wird genauso denken, seine Daten mit einem Backup-Kopierprogramm in regelmäßigen Abständen sichern und sich vernünftigerweise von seinen Originaldisketten Sicherheitskopien ziehen, um die Originale vor unbeabsichtigter Fehlbehandlung zu schützen.

Nur wird sich mancher über die möglichen Schwierigkeiten jedesmal schwarzgeärgert haben, wenn sich herausstellt, daß ein Kopieren der Originaldiskette aufgrund eines dort vorhandenen Kopierschutzes nicht möglich ist. Gerade hier, wo es am wichtigsten wäre, wird der Sinn des Backups in grundlegender Weise zunichte gemacht. Nach Murphys Gesetz des Unheils wird ein Datenverlust hier am ehesten zuschlagen.

Fast jedes Softwarehaus bietet in solchen Fällen zwar den Service des Diskettenumtausches an, doch kann das in keinem Fall den möglichen finanziellen Verlust auffangen, der durch Verzicht auf eine bestimmte Anwendung aufkommen wird. Man stelle sich nur vor, das erworbene Softwareprodukt stamme aus den Vereinigten Staaten und besitzt in der Bundesrepublik keinen direkten Vertrieb. Wie viele Wochen Sie dann für Ihre Umtauschaktion in Kauf nehmen müssen, wieviel Zeit für Schriftverkehr aufgebracht werden muß, malen Sie sich am besten selbst aus. In jedem Fall ist das Problem des Kopierschutzes präsent, und Aussicht auf aufkommende Vernunft auf diesem Gebiet besteht wenig, deshalb die entscheidende Frage: Wie funktioniert so ein Kopierschutz eigentlich? Sind es nicht alles nur Daten, die der Floppycontroller (FDC) mühelos lesen müßte? Um diese und noch weitere Fragen ausführlich zu behandeln, steigen wir ein in den Zug der Kopierschutzmechanismen.

Als Kopierschutz könnte man bereits wilde Farbtabellen auffassen, bei denen man aus einer Farbkombination anhand einer Tabelle eine Ziffernfolge ermitteln und eingeben muß.

Auch gibt es spezielle Zerrbrillen, die einen verzerrt dargestellten Code sichtbar machen, um diesen dann einzugeben. Im Grunde handelt es sich dabei aber nur um eine Art Benutzeridentifikation, und der bleibt ja bei einer Sicherheitskopie derselbe...

Die ersten Kopierschutzmechanismen gab es bereits in den Anfängen der Computerei. Kopierprogramme waren Mangelware, und so reichte es meistens, wenn sich das Programm nach dem Laden automatisch startete und die Break-Taste sperrte. Aber es fanden sich bald findige Freaks, denen es gelang, den Autostart zu überlisten und damit den Kopierschutz zu umgehen.

Am Anfang war die Kassette

Natürlich blieb die Entwicklung nicht stehen, und als die ersten Maschinenprogramme auf den Markt kamen, hatte man es erheblich schwerer. Zum Teil waren die Programme auch so lang, daß es unmöglich wurde, einen Maschinensprache-Monitor zum Ansehen der Programme gleichzeitig im Speicher zu halten. Einige Softwarefirmen setzten auch auf sogenannte Quick-Load-Systeme. Bei diesen Systemen wurde zuerst ein Lader, der selbstverständlich kodiert war, gestartet, der dann erst das eigentliche Programm in den Speicher holte. Dieses Nachladen ging dann verhältnismäßig schnell, da die Baudrate (das heißt, die Übertragungsgeschwindigkeit der Daten von der Kassette in den Computer) ein Vielfaches der normalen betrug. Kopierprogramme, die es mittlerweile auch gab, schafften das natürlich nicht mehr; schließlich handelte es sich ja hierbei um ein völlig anderes Aufzeichnungsverfahren. So hatte man nur die Möglichkeit, sich mit einer Analogkopie zu behelfen. Dazu behandelte man die Datenkassetten einfach wie Musikkassetten und überspielte sie mit einem guten Kassettenrecorder; ein einfaches aber wirkungsvolles Verfahren. Der Nachteil hierbei war nur, daß eine Analogkopie natürlich nicht besser wurde als das Original. Nach mehrfachen Kopiervorgängen erhöhte sich die Fehlerhäufigkeit am schnellsten.

Es war schon eine recht interessante Zeit, und die Softwarefirmen ließen sich immer bessere Schutzmechanismen einfallen; schließlich gab es unter den Freaks sogenannte Cracker, denen es immer wieder gelang, eben diese Schutzmechanismen zu knacken. Teilweise wurden sogar extra Kopierprogramme geschrieben, die sich nur gegen einen bestimmten Schutz richteten. Andere wiederum machten sich an die Arbeit, das komplette Ladeprogramm zu disassemblieren, um an geeigneter Stelle eine Save-Routine einzubauen. Es entbrannte ein endloser Wettlauf zwischen den Softwarefirmen und den Crackern... Aber die Tage der Kassette sind gezählt. Heute benutzt man Disketten, und da treten ganz andere "Ferkeleien" auf.

Daten im Direktzugriff

In den Anfängen der Diskettenkopierprogramme war es für die Softwarefirmen noch einfach, eine schnelle Lösung zu finden: Sie ließen einen Track auf der Diskette unformatiert. Das Resultat war, daß die gängigen Kopierprogramme einen Lesefehler fest- und danach die Arbeit einstellten. Für damalige Verhältnisse einfach und wirkungsvoll. Der Gegenschlag von Firmen, die Kopierprogramme vertrieben, sollte nicht lange auf sich warten lassen: Die nächste Generation konnte durch den User dazu veranlaßt werden, nicht nur von Track 0 bis Track n, sondern auch gewissen Tracks zu überspringen bzw. zu kopieren. Nach diesem wohl ersten Schlagabtausch verfielen Softwareschützer und -entschützer in einen offenen Kampf, der auch heute noch längst nicht beendet ist.

Wie funktioniert's?

Im Grunde verfahren alle Softwarekopierschutzmechanismen nach dem gleichen Prinzip: Entweder werden Programminformationen so auf der Diskette gespeichert, daß sie von einem Kopierprogramm nicht übertragen werden können, oder es werden zum Programm noch zusätzliche Sektoren abgespeichert, die, vom Kopierprogramm nicht übertragen, später vom geschützten Programm aber geprüft werden. Im ersten Fall wird das Programm ohnehin nicht laufen, da durch diesen Schutz Programmteile fehlen. Bei der zweiten Methode ist das Programm zwar vollständig, erkennt aber hierbei selbst, daß es sich um eine Kopie handelt und wird nicht ordnungsgemäß laufen oder zumindest ordentlich abstürzen. Ganz gerissene Programme laufen jedoch trotzdem, nur löschen sie dem ahnungslosen User z.B. die Datendiskette von seinem Textverarbeitungsprogramm oder löschen dieses oder gar beides. Bei anderen gibt es dann erstaunlicherweise keine Möglichkeit mehr, die mühsam erstellten Texte auszudrucken, oder die Druckertreiber sind urplötzlich verschwunden, oder bei einem Adventure kommt der Abenteurer nicht mehr aus einer Fallgrube heraus. Der Fantasie sind da keine Grenzen gesetzt. Einige sehr findige Schützer hatten dann herausgefunden, daß ihre Laufwerke mindestens einen Track mehr nach innen formatieren konnten. Warum also nicht einfach einen Teil des Programms auf diesem zusätzlichen Track abspeichern? Alle Kopierprogramme, die es bis dato gab, duplizierten ja nur die bekannten Tracks. Der zusätzliche Track würde nicht kopiert und das Programm nicht laufen. Jedoch der Haken war, daß einige Anwender Laufwerke besaßen, die diesen Track selbst nicht erreichen konnten. Der Ärger war verständlicherweise groß. Aber auch hier schlugen die Hersteller von Kopierprogrammen zurück: Inzwischen können sie auch Tracks kopieren, die jenseits der Laufwerksspezifikationen liegen.

Die Suche nach dem Anfang

Damit ein Kopierprogramm überhaupt einen Track kopieren kann, muß es dessen Anfang finden. Das liest sich leicht, nur in der harten Welt der Kopierschutzmechanismen ist dies manchmal nicht so: Es können nämlich die Bytes, an denen der Anfang eines Tracks erkannt wird, durch andere ersetzt werden. Ein Kopierprogramm kann den Track nicht mehr erkennen und läßt den Anwender dann alt aussehen. Dies trifft natürlich auch auf die einzelnen Sektoren gleichfalls zu. Die Sektornummern müssen auch nicht einmalig sein. Ein paar doppelte können das Kopierprogramm dann schon streiken lassen. Ein wesentlich "billigerer" Kopierschutz besteht in dem Abprüfen des Schreibschutzes. Da Originaldisketten meistens ungelocht sind, können diese durch das eigentliche Programm schnell überprüft werden. Ist der Schreibschutz nicht vorhanden, so kann es sich bei der eingelegten Diskette um kein Original handeln. Ergebnis: Das Programm erkennt, daß es eine Kopie ist und formatiert sich selber, die Festplatte, die Diskette in einem anderen Laufwerk usw.

Und bist du nicht willig...

Komplizierter ist da schon der echte hardwaremäßige Fehler. Dazu wird eine normal formatierte Diskette einfach an einer Stelle mit einer Nadel zerstört. Danach wird der so erzeugte Fehler exakt bestimmt und kann später von dem geschützten Programm abgefragt werden. Ein Kopierprogramm wird an diesem hardwaremäßigen Schutz seine wahre Freude haben... Disketten benötigen auch Füllbytes, um die einzelnen Tracks der Lauf-

werksgeschwindigkeit anzupassen. Bei der Herstellung der Originaldisketten wird dann ein geschriebener Track nochmal eingelesen, die geschriebenen Bytes werden gezählt und auf einer Stelle der Diskette vermerkt. Da es unwahrscheinlich ist, daß das Laufwerk des Anwenders die gleiche Umdrehungsgeschwindigkeit wie das Originallaufwerk hat, stimmen die geschriebenen Bytes je Track nicht überein. Das Programm kann nun durch Vergleich der geschriebenen Bytes mit der vermerkten Anzahl erkennen, ob es sich bei der Diskette um ein Original oder eine Kopie handelt. Wie man bis hierher unschwer feststellen kann, sind der Fantasie in Sachen Kopierschutz keine Grenzen gesetzt. Nur der FDC muß damit fertigwerden. Ganz besonders effektiv ist der Kopierschutz natürlich dann, wenn der FDC zwar alle Daten einwandfrei lesen kann, beim Schreiben aber seine Arbeit verweigert.

Grenzenlos

Neben dem im CPC und vielen PCs eingesetzten FDC uPD765 gibt es selbstverständlich auch noch andere (z.B. den WD2797). Andere, die zum Teil wesentlich leistungsfähiger sind, ganz besonders im Schreiben und Lesen. So verfügt der uPD765 zwar über den Befehl "kompletten Track lesen", jedoch funktioniert dieser nicht so, wie man sich das wünscht: Daß zwischen den Sektoren zahlreiche Informationen für den FDC stehen, dürfte Ihnen nach der Lektüre unseres Floppy-Kurses kein Geheimnis mehr sein. Das sind Daten, die der FDC unbedingt braucht, um mit der Diskette etwas anfangen zu können. Beim Formatieren einer Diskette werden sie dort hingeschrieben. Wenn Sie jetzt fragen, wo da das Problem liegt, sollten Sie abwarten, denn jetzt kommt's: Im Gegensatz zum uPD765 kann der WD2797 eine komplette Spur schreiben und lesen. Das Lesen ist zunächst einmal uninteressant, wogegen beim Schreiben fast beliebige Bytes auf die Spur gebracht werden können. Man muß als Softwareschützer also nur wissen, welche Daten der uPD765 nicht mag, und schon hat man einen perfekten Kopierschutz.

So funktioniert's

Nach so viel grauer Theorie soll nun die Praxis nicht nachstehen. Selbstverständlich werden Sie auf die Vorzüge des WD2797 verzichten müssen. Wie gehe ich vor? Da der grenzenlosen Fantasie der FDC entgegensteht, kommen wir nicht umhin, uns auf die Fähigkeiten des Formatier-Befehls zu beschränken. Der Formatier-Befehl dient, wie sollte es auch anders sein, zum Formatieren eines gesamten Tracks. Dabei muß der Schreib/Lese-Kopf bereits auf der entsprechenden Spur stehen, da der FDC den Kopf beim Formatieren nicht mehr positioniert. Hier könnte man z.B. schon ansetzen, indem man die Tracks nicht von 0 an aufsteigend, sondern kreuz und quer über die Diskette verteilt formatiert. Ein billiger Trick, der aber nicht unbedingt ein Problem für das Kopierprogramm darstellt. Die letzte Möglichkeit, die jetzt noch bleibt, ist, an den Daten für die Sektoridentifikation zu "drehen". Der Controller verlangt hierfür 4 Bytes: Tracknummer, Sektornummer, Kopfnummer und Sektorgröße. Dabei handelt es sich hier um die Bytes, die direkt auf die Diskette geschrieben werden und zum späteren Identifizieren des Sektors dienen.

Die Möglichkeiten im einzelnen:

Tracknummer: Hier kann man z.B. eine andere angeben, als die physikalische, auf der der Kopf gerade steht. Auch muß sie ja nicht für jeden Sektor innerhalb des Tracks gleich sein.

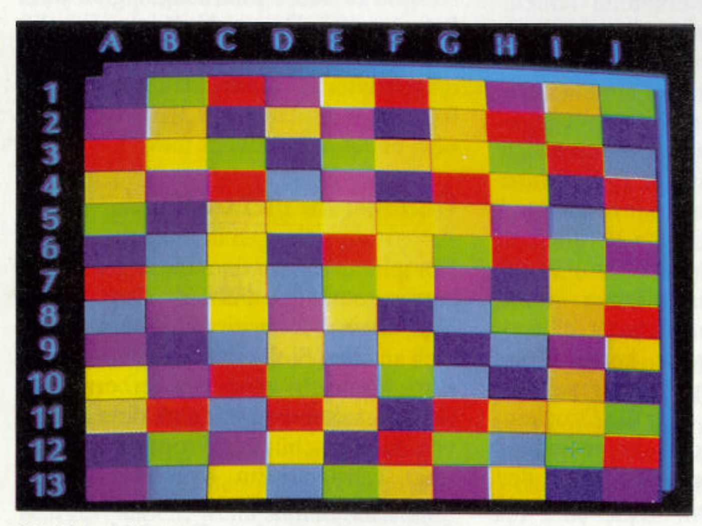
Sektornummer: Sektornummern können, wie bereits erwähnt, mehrfach vergeben werden. Auf diese Weise erreicht man zum einen, daß nie sichergestellt ist, welcher Sektor jetzt gelesen wird, und zum anderen könnte das Kopierprogramm dadurch verwirrt werden.

Kopfnummer: Die Kopfnummer ist bei einseitigen Laufwerken immer Null. Da diese Information aber nur zur Identifizierung dient und nicht die Adressierung im Laufwerk steuert, würde sich hier eine Kopfnummer von 1 bereits als Kopierschutz anbieten.

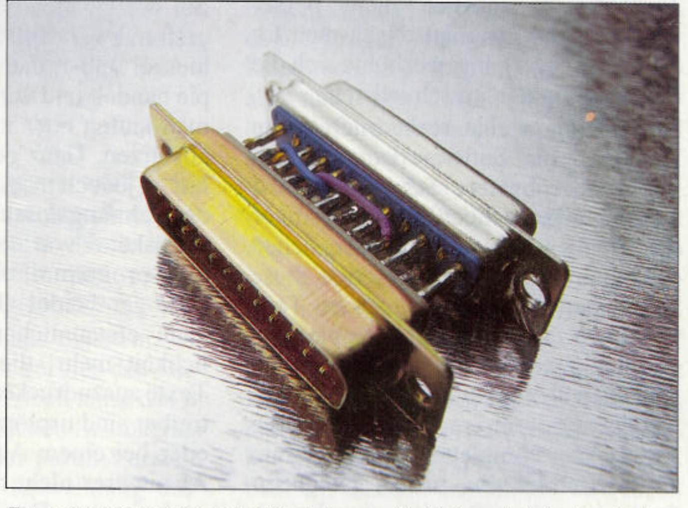
Sektorgröße: Dieser Parameter gibt die Sektorgröße an und kann erhebliche Probleme bereiten, wenn er nicht mit der tatsächlichen Sektorgröße übereinstimmt. Wohlgemerkt, bei diesen Möglichkeiten handelt es sich nur um Tricks, an denen ein Kopierprogramm eventuell scheitern könnte. Es ist durchaus nicht auszuschließen, daß es mittlerweile Kopierprogramme gibt, die (fast) alle Schwierigkeiten meistern. Wirkungslos erscheint dabei auch die Möglichkeit, mit dem Befehl "gelöschte Daten schreiben", einen Sektor als gelöscht zu markieren.

Harte Sachen

Vielversprechender sind da schon die sogenannten Hardwareschutzmecha-



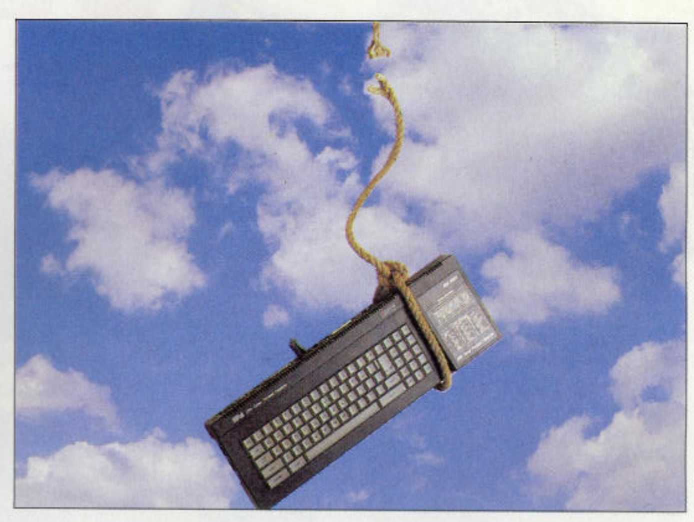
Ein Beispiel für externen Kopierschutz ist eine Farbtabelle, aus der zu Beginn eines Programms Daten abgefragt werden



Ein sogenannter Dongle verhindert zwar die Notwendigkeit des direkten Kopierschutzes auf Datenträger, führt jedoch zu Schwierigkeiten, falls die Ports anderweitig benötigt werden



Auch Disketten gehen irgendwann den Weg alles Irdischen. Peinlich, wenn es ein nichtreproduzierbares Original war: Wartezeit und Ärger sind dann vorprogrammiert



Ein schlecht gemachter Kopierschutz kann unter Umständen zu bösartigen Rechnerabstürzen führen. Eine Schikane für jeden rechtmäßigen Anwender

nismen, im allgemeinen als Dongles bekannt. Ein Dongle ist ein verschweißtes Modul, welches, von außen an den Computer angeschlossen (z.B. über einen Userport), vom Programm abgefragt wird. Als Ausführungen gibt es einfache, die beim Auslesen immer ein bestimmtes Bitmuster ausgeben oder auch solche, die sich programmieren lassen und so nichtreproduzierbare Daten liefern. Den Algorithmus kennt natürlich nur der Programmierer. Eine weitere Möglichkeit besteht darin, daß ganze Programmteile dorthin ausgelagert werden. Ist das Ganze dann noch mit einem guten Softwareschutz versehen, dürfte das Kopieren extrem schwer werden. Da diese Dongles verschweißt sind, so daß man sie beim Öffnen zerstören würde, kann man auch den Nachbau getrost vergessen. Selbst wenn es gelänge, sie unbeschadet zu öffnen, so würde das dem Nachbauer nicht allzu viel nützen, da sie meistens hochintegrierte Schaltkreise enthalten, die ihrerseits nicht im freien Handel erhältlich sind. Andererseits werden auch sogenannte "Programmierbare Gatteranordnungen" verwendet, die es bereits für wenig Geld zu kaufen gibt. Der Nachteil besteht nur darin, daß diese ICs vor dem Auslesen geschützt werden können, so daß ein Knacken völlig unmöglich gemacht wird. Aber so schön die Dongles für die Softwarefirmen auch sein mögen, für die User sind sie äußerst lästig. Belegen sie doch meistens einen Port oder müssen, bei der Verwendung von mehreren Programmen mit Dongles, häufig gewechselt werden. Ein Vorgang, den die Steckkontakte am Userport nicht lange überleben werden.

Viel Geld für nichts

Wenn man sich nun die ganze Geschichte des Kopierschutzes vor Augen hält, wird man unweigerlich zu folgendem Schluß kommen:

1. Die Entwicklung eines Kopierschutzes kostet je nach Aufwand Zeit und damit Geld. Geld, welches natürlich auf den Preis des Produktes aufgeschlagen wird und so letztendlich vom Anwender getragen werden muß, obwohl dieser den Kopierschutz gar nicht wünscht.

2. Bis heute ist jeder Kopierschutz mehr oder weniger geknackt worden. Auch wenn es für die Cracker, nicht zuletzt aufgrund der Dongles, immer schwerer wird, ein Programm zu knacken, so werden sie, allein schon der Herausforderung wegen, immer am Ball bleiben. Das Nachsehen hat hier wieder einmal der normale Anwender, der eben nicht so mir nichts dir nichts ein geschütztes Programm kopieren kann (zu Backupzwecken versteht sich).

Die Firma Apple hat in dieser Sache bereits reagiert und empfiehlt den Softwarehäusern, aus den genannten Gründen auf einen Kopierschutz zu verzichten. Dieser Empfehlung kann man sich besten Gewissens nur anschließen und sie um eine weitere ergänzen. Falls es sich irgendwie vermeiden läßt, lassen Sie kopiergeschützte Software in den Regalen stehen. Programme, die sich eines der Anleitung zu entnehmenden Codewortes bedienen, eine bereits zu Beginn erwähnte Farbschablone benutzen oder andere externe Schutzvorrichtungen besitzen, treffen ins Schwarze, wenn es um das Vermeiden der Kopierwut mancher Zeitgenossen geht. Kopiergeschützte Programme dagegen sind nur teurer und umständlicher in der Handhabung. Der Weg weg von dieser Art des Kopierschutzes ist auf jeden Fall ein Schritt in die richtige Richtung, dem andere hoffentlich folgen werden.

(Marco Pries/jf)

Stichwortverzeichnis der Kopierschutzmechanismen

- Autostart und Sperren der Breaktaste
- Quick-Load-System mit codiertem Ladeprogramm
- Variieren der Baudrate
- Nichtformatieren bestimmter Tracks
- Unregelmäßige Numerierung der Tracks
- Nutzung eines zusätzlichen, normaler Weise ungenutzten Tracks
- Prüfung auf Vorhandensein des Schreibschutzes
- Manipulation der Sektoridentifikation
- Gezielte Zerstörung an bestimmten Stellen des Datenträgers
- Steckmodule (Dongles)

DATABOX

Das ist die Software zur PC International jeden Monat neu

DATABOX:

- mehr als der übliche Softwareservice
- bringt ergänzend sämtliche Listings der jeweiligen Zeitschrift und alle Programmbeispiele auf Kassette oder auf 3-Zoll-Diskette.
- Programme sind, soweit systembedingt möglich, auf allen drei CPC-Modellen lauffähig. Einzelheiten entnehmen Sie bitte der nebenstehenden Aufstellung.
- Soweit die Programme nicht Bestandteil einer Serie sind, befinden sich alle Programme als »ready to run« auf der DATABOX.
- erscheint jeden Monat und trägt das Titelbild des gleichzeitig erscheinenden Heftes.
- Der Datenträger zur PC International enthält außerdem jedesmal ein zusätzliches Bonusprogramm, das nicht im Heft abgedruckt ist.

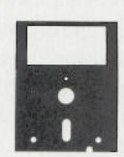
Für alle CPCs als Kassette und 3-Zoll-Diskette. Auch als Abonnement mit Preisvorteil erhältlich

DATABOX HEFT 2/90

CPC	464	664	6128
INHALT:			
Verwaltungshilfe	•	•	•
Sly Sprite	•	•	•
1-kByte-Casino	•	•	•
1-kByte-Fast-Clear	•	•	•
1-kByte-Miniscreen	•	•	•
1-kByte-Multicol	•	•	•
1-kByte-Sternengrafik	•	•	•
1-kByte-Inka	•	•	•
dBase-Listing			
Rolle vor- und rückwärts			
Ein Drucker lernt schreiben			
Checksummer			
Emulator			
Bonusprogramm: Space-Shooting			

Die DATABOX enthält noch Info-Files (*.INF)

Space-Shooting: Weltraumspiel mit viel Action. 50 Spielstufen!




Einzelbezugspreise für Databox:

3-Zoll-Diskette

Inland:		Ausland:	
Einzelpreis	24,– DM	Einzelpreis	24,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**28,– DM**	**Endpreis**	**30,– DM**

Kassette

Inland:		Ausland:	
Einzelpreis	14,– DM	Einzelpreis	14,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**18,– DM**	**Endpreis**	**20,– DM**

Zahlungsweise:
Am einfachsten per Vorkasse (Verrechnungsscheck) oder als Nachnahme zuzüglich der Nachnahmegebühr. Bei Lieferungen ins Ausland ist Nachnahme nicht möglich.

Preisvorteil durch DATABOX-Abo:
Unser beliebter DATABOX-Service kann selbstverständlich auch im Abonnement bezogen werden. Dadurch sparen Sie Mühe und haben außerdem noch einen Preisvorteil gegenüber dem Einzelbezug.

Das DATABOX-Abo kostet:

Als Kassette für 1/2 Jahr (6 Lieferungen):
Im Inland und West-Berlin.................90,– DM
Im europäischen Ausland.................100,– DM
Im außereuropäischen Ausland.........120,– DM

Als Diskette für 1/2 Jahr (6 Lieferungen):
Im Inland und West-Berlin................150,– DM
Im europäischen Ausland.................160,– DM
Im außereuropäischen Ausland.........180,– DM

Als Kassette für 1 Jahr (12 Lieferungen):
Im Inland und West-Berlin................180,– DM
Im europäischen Ausland.................200,– DM
Im außereuropäischen Ausland.........240,– DM

Als Diskette für 1 Jahr (12 Lieferungen):
Im Inland und West-Berlin................300,– DM
Im europäischen Ausland.................320,– DM
Im außereuropäischen Ausland.........360,– DM

Bitte benutzen Sie für Ihre Bestellung die Abo-Karte.

Widerrufsrecht: Jeder Abonnent hat das Recht, seine Bestellung innerhalb einer Woche beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich zu widerrufen. Die rechtzeitige Absendung des Widerrufsschreibens genügt zur Fristwahrung.

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

Epyx Action

Viel Spiel fürs Geld

Compilations sind eine feine Sache. Wenn man fünf noch dazu gute Spiele für den Preis von einem einzigen bekommen kann, dann lohnt es sich auf jeden Fall zuzugreifen.

Bei der hier vorliegenden Zusammenstellung handelt es sich um eine Sammlung von – im weitesten Sinne des Wortes – Sportsimulationen. Basketball, Autorennen, Winter- und Strandspiele sowie ein Action-Klassiker dürften für lange und spannende Abende sorgen. Im einzelnen sind dies:

California Games

Am Strand von Kalifornien gibt es ganz besondere Sportarten. Diese sechs Disziplinen sind auf keiner Olympiade zu finden, trotzdem machen sie einen Heidenspaß.
Wellenreiten und Frisbee sind hinlänglich bekannt, obwohl die dabei möglichen Tricks nicht unbedingt zum Allgemeinkönnen gehören.
Aber was dem Spieler in der Skateboardröhre abverlangt wird, das erfordert schon eine ganze Menge Übung. Handstand, Drehung in der Luft und andere Kunststücke sind in Perfektion hinzulegen, um möglichst viele Punkte einzuheimsen.
Um sich von den vielen Stürzen etwas zu erholen, ist Foot Bag die richtige Disziplin. Bei dieser etwas exotisch anmutenden Sportart geht es darum, einen kleinen Ball wie ein Jongleur eine Minute 15 Sekunden in der Luft zu halten, ohne die Hände dabei zu benutzen. Ellenbogen, Kopf, Knie, Füße und Hacken, mit etwas Gelenkigkeit schafft man dabei auch noch die irrsten Verrenkungen, um das Punktekonto weiter aufzufüllen. Es macht eine Menge Spaß und ist gar nicht so schwer.
Rollschuhlaufen hat ja wohl jeder gelernt und so ein paar kleine Hindernisse zu überspringen, dabei noch eine Drehung zu vollführen und das Ganze in einer Superzeit zu Ende zu bringen, ist doch wohl kein Problem, oder?
Extreme Sturzgefahr herrscht auch beim BMX-Fahren. Der Kurs ist mit gemeinen Hindernissen gespickt, so daß schon einfaches Geradeausfahren zum Glücksspiel wird. Da auch hier, wie in allen Disziplinen, gegen ein Zeitlimit gefahren werden muß, ist viel Übung nötig, um dabei durch ein paar Kunststückchen, wie Sprünge, Drehungen und anderes mehr, Punkte zu scheffeln.

4x4 Off Road Racing

Markige Geländewagen, gute Zusatzausrüstungen, harte Pisten und noch härtere Gegner versprechen viel Action bei diesem Autorennen quer durch die Pampa. Vier Autos, die nach Geschmack aufgerüstet werden können, stehen zur Auswahl und ebenso vier Pisten, von der staubigen Baja California, der Schlammschlacht in Georgia bis zur Eisbahn an der kanadischen Grenze. Egal welche man sich davon aussucht, Mensch und Maschine werden bis an die Grenze ihrer Leistungsfähigkeit beansprucht. Schlaglöcher, Felsbrocken, Flüsse, die man nur dann ohne Zeitverlust überspringen kann, wenn man die Sprungschanze am Ufer genau trifft, Hitze und Kakteen (auch im hohen Norden) machen den Fahrern zu schaffen. Und wehe dem, der nicht die nötigen Mittel zur Reparatur mitgenommen hat! Er wird von seinen sechzehn Kontrahenten mit höhnischem Hupen überholt.
Bei diesem Camel-Trophy-Verschnitt heißt es: Nimm die beste Kiste, die Du finden kannst, dann fahr' drauflos und lasse Deine Skrupel zu Hause! Denn nur einer kann gewinnen und wenn Du die anderen nicht von der Piste fegst, wird einer von ihnen es sicherlich mit Dir tun.

The Games – Winter Edition

Etwas mehr Fairness herrscht bei den olympischen Winterspielen, obwohl es auch hier heißt: "Dabeisein ist alles, gewinnen ist mehr". Die Olympiade ist ein richtiges Familien- bzw. Gruppenspiel, das umso mehr Spaß macht, je mehr Personen (bis zu acht können es werden) an den Wettkämpfen teilnehmen.
Im ersten Menü hat man die Wahl zwischen Übungsläufen oder Spiel, wobei etwas Training nie schaden kann. Das ist schon deshalb nötig, damit man später mit der Steuerung zurechtkommt und nicht dauernd auf die Nase fällt.
Nach einigen Übungsrunden beginnt dann der Ernst des Lebens. Sieben Disziplinen stehen zur Auswahl. Man kann sie einzeln, alle nacheinander oder eine Auswahl davon spielen. Danach werden die Namen eingegeben, die Nationalitäten und Hymnen ausgesucht, und schon geht es los mit der feierlichen Eröffnungszeremonie und dem ersten Wettkampf.
Beim Rodeln kommt es darauf an, sich gleich zu Beginn ein ordentliches Tempo zu verschaffen und dann, ohne zu viel anzuecken, eine der vier möglichen Bahnen hinunterzujagen.
Beim Langlauf stehen drei verschieden lange Strecken zur Auswahl, an deren Ende Läufer wie Joystick ziemlich an Muskelkater leiden dürften. Den Langläufer dabei so richtig in Schwung zu bringen, erfordert gutes Timing und genaues Beobachten der Bewegungsabläufe.
Dasselbe gilt für den Eisschnellauf, bei dem ein ausgedehntes Trainingsprogramm empfehlenswert ist, da sich der Läufer sonst mehr als Eisbrecher betätigt, als daß er auf einer der vier Bahnen Weltrekorde aufstellt.



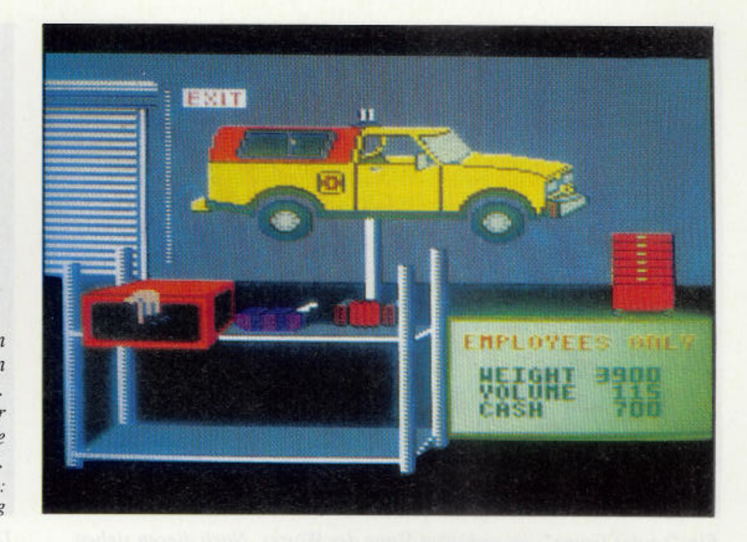
Pralle Sonne, heißer Strand und lockere Spielchen, das läßt man sich besonders jetzt im Winter gefallen
Screenshot: California Games

Vor den Erfolg im Eiskunstlauf haben die (olympischen) Götter die Musik gesetzt. Drei Stücke stehen zur Verfügung, zu denen eine Darbietung aus acht Sprung- und Tanzelementen zusammengestellt werden muß.

Beim Slalom ist das alles viel einfacher. Links und rechts um die Tore der vier Streckenvorschläge herumfahren kann doch jeder, oder etwa nicht?

Der Abfahrtslauf hält einige Bonbons bereit. Nicht nur, daß man wie im richtigen Leben auch in der Hocke den Hang hinunterjagen kann, auch Luftsprünge gibt es sowie die Möglichkeit, an schwierigen Stellen Kameras aufzustellen, die den Kurs aus unterschiedlichen Perspektiven zeigen. So dürfte es eigentlich keine Schwierigkeiten bereiten, die Piste in Rekordzeit zu meistern. Gips wird allerdings nicht mitgeliefert, sondern ist im Baumarkt zu erwerben.

Und dann gibt es noch das Skispringen. Die richtige Kombination aus Tempo, Timing und Körperbeherrschung ist Voraussetzung, von der Schanze überhaupt (im Flug und nicht im Sturz) herunterzukommen. Wenn dann auch Haltung und Landung noch stimmen, ist eine Höchstpunktzahl vorprogrammiert, die sich in der Liste der Weltrekorde sicherlich hübsch macht!

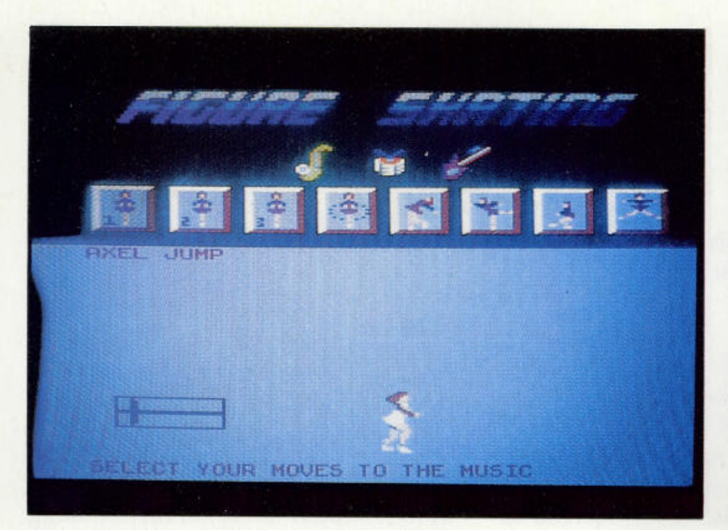


Aus dem trockenen Staub direkt in den matschigen Schlamm. Ein Wechselbad der Gefühle erwartet Sie bei dieser Rallye.
Screenshot: 4X4 Off Road Racing

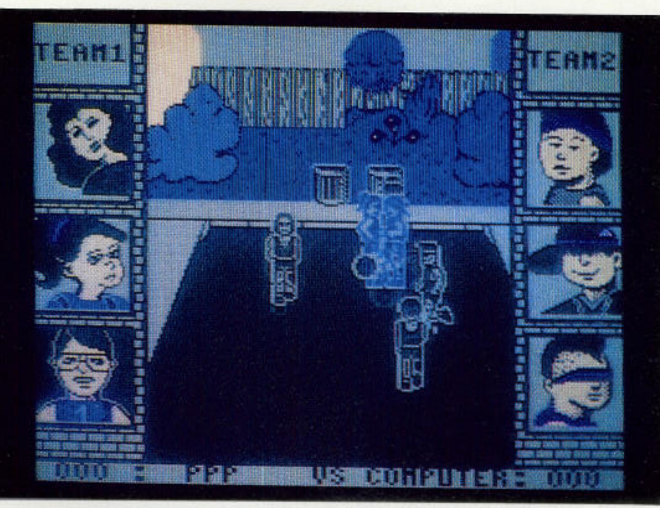
Am Ende jeder Wettkampfdisziplin steht natürlich die Siegerehrung samt Nationalhymne auf dem Programm, wie auch am Ende der gesamten Olympiade eine Schlußfeier nicht fehlen darf. Und wem die Zeit bis zu den nächsten olympischen Spielen zu lang ist, der kann sich mit diesem Programm stilvoll die Zeit vertreiben.

Impossible Mission II

Um die Welt vor dem verrückten Professor Elvin zu retten, rennt unser Held im Dauerlauf durch die Gänge eines unterirdischen Hoch- (oder besser Tief-) Hauses, dessen Stockwerke per Fahrstuhl miteinander verbunden sind.



Ein "cooles Game", im wahrsten Sinne des Wortes. Nach diesen sieben Wintersportdisziplinen kann ein heißer Grog nicht schaden
Screenshot: The Games Winter Edition



Die Bälle werfen, bis des Nachbars Scheibe vor Freude zerspringt: Basketball im Hinterhof
Screenshot: Street Sports Basketball

Dabei arbeitet nicht nur die Zeit gegen ihn, auch Horden von mehr oder weniger tödlichen Robotern behindern ihn ständig bei seiner Arbeit. Diese besteht darin, die Sicherheitssysteme der Festung zu unterwandern und Losungszahlen und -melodien zu finden. Sie sind in den Gegenständen der mit Robotern verseuchten Räume bzw. in Tresoren versteckt, die nur durch Sprengung zu öffnen sind. Ziel der Aktion ist, den Hauptaufzug zu aktivieren, bis in Elvins Kommandozentrale vorzudringen und die Weltzerstörungsschaltung (tolles Wort) außer Betrieb zu setzen. Hört sich einfach an, ist es aber nicht, auch wenn man den lästigen Gegnern mit einem eleganten Salto vorwärts immer wieder entkommen kann.

Langeweile kommt auch nach häufigem Spielen nicht auf: Bei jedem neuen Spiel ist die Lage der Räume anders. Und da sowohl die Losungszahlen wie auch die zum Öffnen der Durchgänge nötigen Melodiefetzen immer wieder unterschiedlich sind und so jedes Spiel anders ausfällt, ist für lang anhaltende Motivation gesorgt.

Bei Impossible Mission II handelt es sich um ein klassisches Plattformspiel, an dem sich schon Legionen von Spielern die Zähne ausgebissen haben und an dem sie trotzdem immer noch knabbern. Deshalb sollte dieser Oldie but Goodie in keiner Softwaresammlung fehlen, und wer ihn noch nicht hat – nun, hier ist eine preiswerte Möglichkeit, diese Lücke zu stopfen.

Street Sports Basketball

Ein Spiel, das (fast) keine Wünsche offen läßt, denn so viele Optionen wie hier sind selten: frei wählbare Endstandpunktzahl, Ein- oder Zwei-Spieler-Modus, drei Schwierigkeitsstufen, vier verschiedene und detailreiche Hintergrundgrafiken. Die beiden Teams, die aus je drei Spielern bestehen, können aus zehn Charakteren zusammengestellt werden, die unterschiedliche Talente haben. So kann man eine schlagkräftige Mannschaft bilden, die auf einem Hinterhof, einem Parkplatz, in der Schule oder auf dem Spielplatz zu Hause dem gegnerischen Team zeigt, was eine Harke ist.

Ein Wermutstropfen mischt sich jedoch in dieses ansonsten rundum gelungene Spiel: Die zwei jeweils gerade aktiven Spieler strahlen leider beide in ein und demselben wunderbaren Himmelblau, so daß es ziemlich schwerfällt, bei einem heißen Match festzustellen, in wessen Besitz der Ball gerade ist.

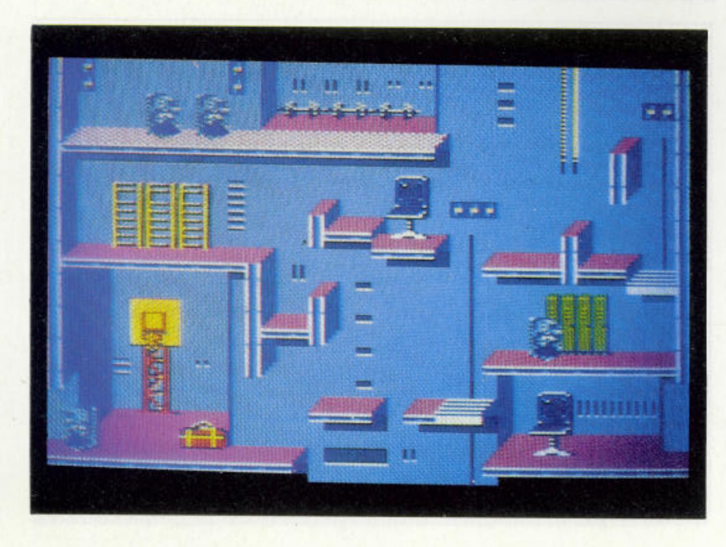
Zusammenfassend...

...wäre wohl zu sagen, daß es sich bei dieser Spielesammlung um eine durchaus lohnende Anschaffung handelt, selbst wenn man das eine oder andere dieser noch gar nicht so alten Programme bereits besitzt.

Bedienungsfreundlichkeit	2	Grafik	2	Endnote
Motivation	2	Sound	4	2

Epyx Action
Hersteller: US Gold
Steuerung: Tastatur/Joystick
Preis: 64,95 DM

(Antje Hink/jf)



Die Erledigung dieses unmöglichen Auftrags bringt ein ständiges Auf und Ab
Screenshot: Impossible Mission II

Fighting Soccer

Ein tolles Titelbild, ein guter Sound, ein olympisches Match, viele Optionen wer gegen wen – das alles kann eine gute Ausgangsbasis für ein erstklassiges Spiel darstellen. Aber ist es hier auch eins geworden?



Hängen wir den Mantel des Schweigens über dieses wenig gelungene Fußballspiel

Auf dem grünen Rasen, von dem immer nur ein Ausschnitt in der Vogelperspektive zu sehen ist und der in den Proportionen leider überhaupt nicht stimmt, tummeln sich jede Menge Beintreter. Die einen im blauen, die anderen im blau-weiß gestreiften Trikot, wodurch es ziemlich schwierig wird, Freund und Feind in der Hitze des Kampfes auseinanderzuhalten.

Laut (spärlicher) Spielanleitung gibt es unter anderem die Möglichkeit, im Zwei-Spieler-Modus mit zwei Joysticks zu spielen. Da der Computer aber selbst bei dieser Option den Gegenpart übernimmt, ist ein echtes Spiel gegen einen menschlichen Gegner trotzdem nicht möglich.

Fighting Soccer ist ein Spiel für diejenigen, die alles kaufen, was an Sportsimulationen auf den Markt kommt. Wer jedoch Fußball liebt, der sollte sich für seinen (halben) Blauen lieber Eintrittskarten zu einem echten Spiel besorgen – der Spielspaß dabei hält länger an, mindestens 90 Minuten, und von so einer langen Spielfreude kann dieses Programm nur träumen!

(Antje Hink/jf)

Bedienungsfreundlichkeit	Grafik	Endnote
4	3	5
Motivation	**Sound**	
5	4	

Fighting Soccer
Hersteller: Activision
Steuerung: Tastatur/Joystick
Preis: 49,95 DM



Assembler von der Pike auf

Hersteller: Sybex Verlag GmbH, Düsseldorf
Vertrieb: DMV-Verlag, Eschwege
Programm: BASIC und Maschinencode
Monitor: Farbe/Grün
Steuerung: Tastatur
Preis: 49,– DM (Buch und 3-Zoll-Diskette, Buch und Kassette 39,– DM)

CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

Der CPC-Assembler-Kurs von Sybex im Test.

"Das Buch führt den Anfänger schrittweise in die Programmierung des Z80 ein. Dabei werden Vorkenntnisse nicht vorausgesetzt." So verspricht es die rückseitige Beschreibung zu diesem Assembler-Kurs. Warten wir ab, ob das Produkt dem Versprechen gerecht wird.

Das Buch...

Zu einem Preis von 49,– DM bekommt man ein 229seitiges Buch und eine 3-Zoll-Diskette mit sieben Programmen. Das Buch ist eine sehr gute Übersetzung aus dem Englischen. Es gibt an vielen Stellen den typischen englischen Sprachgebrauch wieder und führt vom Anfang bis zum Ende einen Dialog mit dem Leser: "Wenn Sie bereits eine Idee entwickelt haben, wie Sie die Zahlen 1 bis 9 auf dem Bildschirm abbilden können, sollten Sie an dieser Stelle das Buch schließen und versuchen, die Aufgabe in eigener Regie zu lösen. Wenn es nicht klappt, ärgern Sie sich bitte nicht. Die Lösung folgt auf dem Fuße." So ähnlich geht es über das ganze Buch hinweg. Einen Grund zum Ärgern gibt es aber wirklich nicht, da der Kurs sehr übersichtlich strukturiert ist und durch viele einfache Beispiele für den Anfänger überaus leicht zu verstehen ist. Begrüßenswert ist auch die Verwendung von englischen Standardbegriffen neben den weniger gebräuchlichen deutschen.



Das Menü vom Assemblerkurs

Gegliedert ist das Buch in zehn Kapitel, die jeweils in sich abgeschlossene Befehlsgruppen erläutern und am Ende Übungsaufgaben sowie Fragen zu dem bisher Erlernten stellen. Dabei ist erwähnenswert, daß das Buch sich nicht über den Leser stellt und trockene Theorie vermittelt, sondern daß alle Beispiele anhand des mitgelieferten Assemblers sofort ausprobiert werden können und auch sollen, wenn man sich an das Buch hält.

Für einen Fortgeschrittenen mag die eine oder andere Vorgehensweise des Buches etwas ungewöhnlich sein. So erwartet man am Anfang doch eher eine Beschreibung, wie man den Assembler startet und nicht eine Rechenaufgabe in Maschinensprache. Doch gerade dieses erweist sich für den absoluten Anfänger als äußerst vorteilhaft. Das Handbuch beschreibt dabei auch die einzelnen Handgriffe, um schließlich den Assembler zu laden; getreu dem Motto "alles zu seiner Zeit".

Der Bonus...

Als Bonus werden dann am Ende des Kurses noch einige nützliche Grafikroutinen geboten, die ihresgleichen suchen und sich sehr leicht als RSX unter BASIC einbinden lassen. So findet man z.B. eine Circle-Routine, die dank des gewählten Bresenham-Algorithmus an Geschwindigkeit wohl nicht zu überbieten ist. Es ist nahezu schon selbstverständlich, daß eine genaue Beschreibung des Algorithmus in einem so guten Buch nicht fehlt.

In diversen Anhängen werden dann noch einmal alle Befehlsgruppen in Tabellenform dargestellt, damit man einen abschließenden Überblick bekommt. Abgerundet wird der Kurs durch einen kurzen Abriß der wichtigsten Betriebssystemroutinen und deren Erklärung sowie eine Einführung in die verschiedenen Zahlensysteme.

Im wahrsten Sinne des Wortes perfekt wäre das Buch, wenn nicht das Thema I/O-Befehle zu kurz geraten wäre. Hier fehlt leider ein tieferer Einschnitt in die Spezialitäten der CPC-I/O und was dabei zu beachten ist. Ein Manko, welches aber durchaus zu verzeihen ist, wenn man bedenkt, daß ein Anfänger wohl kaum mit der Programmierung der I/O-Ports beginnen wird.

...und die Software

Wie bereits erwähnt, befinden sich auf der Diskette sieben Programme. Im einzelnen sind dies der Lader, der Assembler-Basicteil, der Assembler- Maschinenteil, ein Programm zur Darstellung von verschiedenen Zahlensystemen, eine Grafikdemo mit den im Buch enthalten RSXes sowie die Grafikroutinen im Sourcecode.

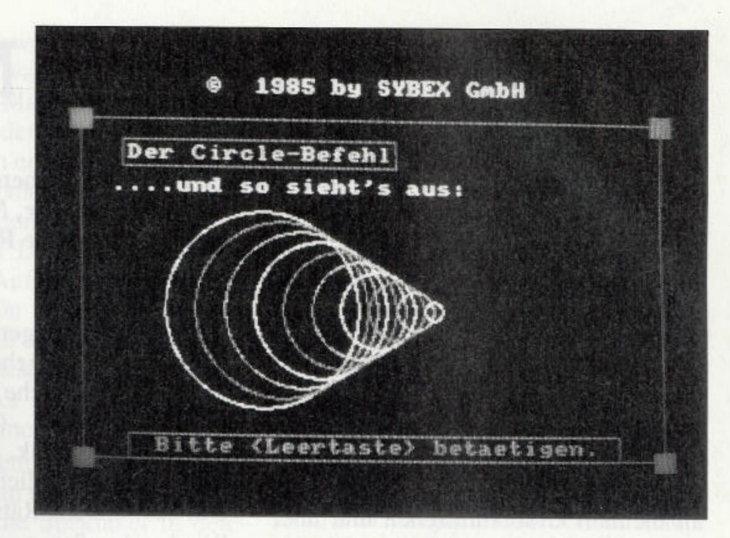
Nach dem Starten meldet sich der Assembler mit einem Menü, wie es in Bild 1 zu sehen ist. Besonders vorteilhaft an diesem Menü ist die Einfachheit, mit der auch Anfänger, die zuvor noch nie mit einem Assembler gearbeitet haben, klarkommen. Diese Menütechnik erstreckt sich über das gesamte Programm, so daß man Schritt für Schritt zum Ziel geführt wird.

Kompromisse muß man allerdings beim Einsatz des Editors in Kauf nehmen. Gerade hier hätte man auf eine bessere Bedienbarkeit achten sollen. So sollte man sich z.B. den Inhalt einer zu editierenen Zeile vor dem Editieren merken. Der Editor erwartet nämlich gleich die neue Zeile, ohne vorher den Inhalt der alten auszugeben. Auch ist das Bildschirmfenster etwas zu klein geraten, man hätte hier doch besser auf

die Copyright-Meldung verzichten sollen. Wünschenswert wäre auch eine Umschaltungsmöglichkeit auf 80 Zeichen pro Zeile für diejenigen, die keinen Farbmonitor benutzen; 40 Zeichen sind für ein Assembler-Programm häufig zu wenig. Wenn man sich aber wieder den Zweck dieses Kurses vor Augen hält, wird einem klar, daß dieser für den Einsteiger gerade richtig ist. Möchte man nach diesem Kurs seine Assemblerkenntnisse vertiefen und größere Programme schreiben, wird man ohnehin auf die Grenzen unter BASIC stoßen und auf CP/M ausweichen müssen.

Zum Abschluß

Um also abschließend die anfängliche Frage zu beantworten: Trotz der erwähnten minimalen Mängel sowohl Buch als auch Software betreffend, ist dieser Assembler-Kurs hervorragend für den interessierten Anfänger, der sich in die Programmierung des Z80-Assemblers einarbeiten will, geeignet. Freilich sollte man gewisse Erfahrungen in der BASIC-Programmierung mitbringen, da eine logische Programmierung in Assembler noch wichtiger ist als in BASIC. Der Grund: Ein Fehler im Maschinencode läßt das ganze Programm zusammenbrechen, und der Computer ist nur noch durch einen Kaltstart wieder ansprechbar.



In der Grafikdemo werden Beispiele gezeigt, die in eigene BASIC-Programme übernommen werden können

Bedienungsfreundlichkeit	2	Grafik	—	Endnote 2
Nutzen	2	Dokumentation	2	

Fazit

Das Produkt ist äußerst empfehlenswert. Mir ist kein besseres dieser Art bekannt.

(Marco Pries/cd)

Gamers Message

In diesem Monat wollen wir uns konzentriert mit dem Abenteuer Scapeghost befassen. Lassen Sie sich in eine Welt voller Geister, Mörder und Dealer entführen, eine Welt, in der Drogen eine nicht geringe Rolle spielen. Für Spannung und Nervenkitzel ist also bestens gesorgt.

Die erste Nacht

Schon mal bei der eigenen Beerdigung zugesehen? Ist schon ein komisches Gefühl, so als quasi neugeborenes Gespenst die Leute zu beobachten, wie sie an meinem Grab rumstehen und über meine Unfähigkeit als Drogenfahnder lästern, den letzten Fall zu einem guten Ende zu bringen. Schließlich kann ich doch nichts dafür, daß ich hinterrücks erschossen wurde und meine Partnerin Sarah den Kerlen in die Hände fiel! Irgend etwas ist komisch an der ganzen Sache, und ich muß herausfinden, was, um Sarah zu retten und meinen guten Ruf wiederherzustellen.

Die abwertenden Kommentare der Polizisten ärgern mich und so sehe ich sie mir etwas genauer an. Zu meiner Überraschung stelle ich fest, daß einer von ihnen gar kein Polizist ist, sondern der Bande angehört, der ich auf der Spur war, bevor jemand meinem Leben so unsanft ein Ende setzte. Leider kann ich dem Kerl keins auf die Nase geben, weil meine Kräfte dazu noch nicht wieder ausreichen. Daher nehme ich mir vor, meine matschigen Muskeln wieder in Form zu bringen. Hanteln gibt's hier nicht, also beginne ich ganz bescheiden mit einem Distelsamen, den ich nördlich von meinem Grab finde.

Ein trockenes Blatt ist das nächste Übungsobjekt, dann die weiße Karte auf meinem Kranz, der Kieselstein und schließlich ein Zweig, der auf einem Komposthaufen liegt. Der Erfolg ist beachtlich.

Durch die Schuppentür, hinter der sicherlich interessante Dinge verborgen sind, kann ich aber leider trotzdem nicht gehen. Deshalb warte ich, bis der Oberfriedhofswärter mit seinem Autochen aufkreuzt und sie öffnet (nachdem er einen verwelkten Kranz auf den Kompost geworfen hat). Drinnen fällt mir sofort eine kleine Glocke ins Auge, für die sich bestimmt ein guter Verwendungszweck finden läßt. Da meine noch immer recht geringen Kräfte völlig verschwinden, solange Leute in meiner Nähe sind, dauert es etwas, bis ich die Glocke nach vielen vergeblichen Versuchen nach draußen bringen und sie dort ablegen kann. Wenn mir noch Zeit bleibt, gehe ich sofort wieder hinein und versuche, auch die Uhr mit hinauszunehmen. (Falls das nicht klappt: keine Panik, es gibt später noch eine zweite Möglichkeit, das mit der Uhr verbundene Rätsel zu lösen.)

Wieder draußen angelangt, nehme ich die Glocke und gehe entweder zu meinem Grab zurück oder warte darauf, daß ich mit Einbruch der Dunkelheit von alleine dorthin transportiert werde.

Treffunkt Grab

Am Grab werde ich von Joe, einem netten Mitgespenst, freundlich begrüßt und zu einer Friedhofsbesichtigung eingeladen. Ich nehme das Angebot an und laufe Joe einfach hinterher. Er führt mich zuerst zu Bert und Edna, die sich auch im Jenseits streiten wie eh und je. Ich höre den beiden fasziniert zu. Bis ich jedoch ihr Problem verstehe, ist Joe schon weitergegangen. Nichts wie hinterher!

Bei David, einem ziemlich eingebildeten Typen, ist der nächste Halt. Ich bringe mal kurz die Urne zum Wackeln, die auf seinem Sarkophag steht, um dem Schnösel zu zeigen, daß er hier nicht der einzige ist, der Muskeln hat. Das imponiert ihm so, daß er tatsächlich mitkommt, als ich ihn dazu auffordere.

Der Colonel, den wir als nächstes aufsuchen, ist damit beschäftigt, jugendliche Randalierer zu verscheuchen und kann uns deshalb leider noch keine Gesellschaft leisten.

Nun besuchen wir ein Gespenst, das in seinem vorigen Leben blind war. Violet traut sich deshalb nicht von ihrem Grab weg, weil sie Angst hat, es nicht wiederzufinden. Ihr zu helfen ist kein Problem: Ich binde die kleine Glocke an einen Busch, der neben ihrem Grab wächst. Schon hat sie einen Orientierungspunkt und mein Gefolge kann bald um ein weiteres Mitglied verstärkt werden.

Bei Edith, die wir nun aufsuchen, liegt der Fall schon etwas schwieriger. Sie glaubt, daß sie durch einen Fluch bis zum Ende der Zeit in diesem abgelegenen Winkel des Friedhofs bleiben muß. Mein erster Gedanke ist: »Die tickt wohl nicht ganz richtig«. Aber dann dämmert mir was – und falls man die Uhr bereits aus dem Schuppen herausgebracht hat, wäre dies der passende Zeitpunkt, sie ihrer Bestimmung zuzuführen...

Ich gehe also die Uhr holen und werfe sie in die Vase auf Ediths Grab, wo sie in tausend Stücke zerspringt. Edith fühlt sich erlöst und schließt sich meinen Mitgeistern an. Auch Violet lade ich nun ein, mir bei meinen weiteren Aktionen zu helfen.

Gespenstisches Gekraule

Irgendwo auf dem Friedhofsgelände treffe ich dann auf einen Hund (goto dog), der verzweifelt versucht, eine Fischgräte loszuwerden, die sich in seinem Hals verhakt hat. Auf Zurufe reagiert er nicht, also versuche ich es mit Streicheln (pat). Vor lauter Panik haut das Tier erstmal ab, aber als ich ihm folge, läßt er sich doch noch ein paarmal den Rücken kraulen, dann traut er mir soweit, daß ich mit meiner Geisterhand in seinen Hals greifen und die Gräte entfernen kann. Das ist für den genervten Hund denn doch zu viel, und er sucht erstmal das Weite. Da ich sicher bin, daß er sich bald beruhigen und wieder zurückkommen wird, wende ich mich anderen Aufgaben zu.

Ich begebe mich zu Ednas altem, verwelkten Kranz und nehme die daran geheftete gelbe Karte mit, die ich dann auf meinen eigenen, noch frischen Kranz lege. Inzwischen ist der Hund (hoffentlich) wieder aufgekreuzt und schnappt sich den Kranz mit der gelben Karte, nachdem ich vergeblich versucht habe, ihn aufzuheben. Ich begebe mich nun zum Grab der Willmots, deren Streit mit Hilfe des vom Hund abgelieferten Kranzes ein Ende gemacht wird. Ich bitte die beiden, mir zu folgen und gehe nach Südosten, wo zwischen vielen Trümmern auf einem Sockel die Statue eines schlafenden Männekens steht. Ihm versetze ich einen sanften Klaps, Alex erwacht zum Leben: Ein Geist mehr, der mir folgt, als ich mich nun nach Süden wende.

Hier steht eine Sonnenuhr herum, die im Dunkeln nur leider keinem nützt. Meine Kräfte reichen noch nicht aus, um das Ding umzuwerfen. Als jedoch

der Hund wieder auftaucht, hilft er mir dabei. Die anderen Geister sind etwas ungehalten, als die Sonnenuhr kaputtgeht, ich jedoch stähle meine Muskeln zum Maximum, indem ich den abgebrochenen Zeiger aufhebe. Nun wird es Zeit, sich um Ediths Problem zu kümmern, falls man das noch nicht konnte. Ich gehe zu diesem Zweck in die nordwestlichste Ecke des Friedhofes, wo ein paar Holunderbüsche wachsen. Unter ihnen verborgen liegt eine Ausgabe des Time-Magazins. Nachdem ich dem Hund gezeigt habe, daß ich das Heft haben will, nimmt er das Heft in die Schnauze und folgt mir zu Edith, die das Magazin voller Wonne vernichtet und anschließend ebenfalls bereit ist, mir zu folgen.

Ghostwork

Das ist auch bitter nötig, denn bei den nun kommenden Aktionen brauche ich jede Hilfe, die ich kriegen kann.
Um die Schuppentür zu öffnen, ist Teamarbeit angesagt. Bert muß dreimal warten, dann am kurzen Hebel ziehen. Edna wartet zweimal, zieht am mittleren Hebel, Edith wartet einmal und zieht am langen Hebel. Ich selbst drehe den Zylinder des Schlosses um, und die Tür ist entriegelt. Nun ziehe ich zusammen mit dem Hund an dem Seilgriff, die Tür geht auf, der Hund stürzt in den Schuppen und verschwindet im Dunkel mit dem Stullenpaket des Friedhofsarbeiters im Maul. Gönnen wir's ihm!
Im Innern des Schuppens befindet sich ein Knopf, der die Außenbeleuchtung regelt. Zusammen mit einem meiner Mitgeister entwickeln wir genug Kraft, um ihn zu betätigen. Als das Licht erlischt, werden die Rowdies draußen nervös. Wir machen das Licht wieder an, die Jungs kriegen Panik, machen sich über die Friedhofsmauer davon, und der Colonel hat ein Problem weniger. Bevor wir alle zusammen mit dem Colonel den Schuppen wieder verlassen, schalten wir die Außenbeleuchtung aber noch endgültig aus, um unseren Bewegungsspielraum etwas zu erweitern.

Die Geister, die ich rief...

Nun gilt es vor allem, die Dealer daran zu hindern, sich die Drogen unter den Nagel zu reißen, die in dem Gewölbe neben der baufälligen Mauer versteckt sind. Dazu ist eine gut abgestimmte Aktion aller neun zur Verfügung stehenden Geister nötig. Ich verteile die Aufgaben: Alex, wait 8, push wall. Joe, wait 7, push wall usw, bis ich als letzter gegen die Mauer drücke. Dem nun aufkommenden gespenstischen Sturm könnte auch eine weniger kaputte Mauer nicht standhalten und unter den herabstürzenden Trümmern werden Gewölbe und Drogen völlig verschüttet. Die Aufräumungsarbeiten werden ganz schön Zeit in Anspruch nehmen, so daß wir uns beruhigt in unsere Gräber zurückziehen und einer neuen Nacht voller Aufregungen entgegenschlafen können.
Nach der schon sehr erfolgreichen ersten Nacht darf man gespannt sein, wie das mitternächtliche Spektakel in den kommenden beiden Nächten zuende geführt wird. Bis dahin lassen wir unsere Geister spuken und widmen uns anderen Dingen.
Eine Reihe interessanter Tips erreichte uns von Ingo Grafe und Joachim Sobczak:

The Real Ghostbusters

Das Programm besitzt einen Cheat-Modus, der außer Unsterblichkeit die Möglichkeit gibt, sich in weitere Levels zu transportieren. Man hat lediglich hintereinander die Anfangsbuchstaben des Titels zu drücken: The Real GhostBusters! Bei jedem folgenden E (Exit) gelangt man nun einen Level weiter.

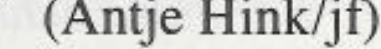
Vermeer

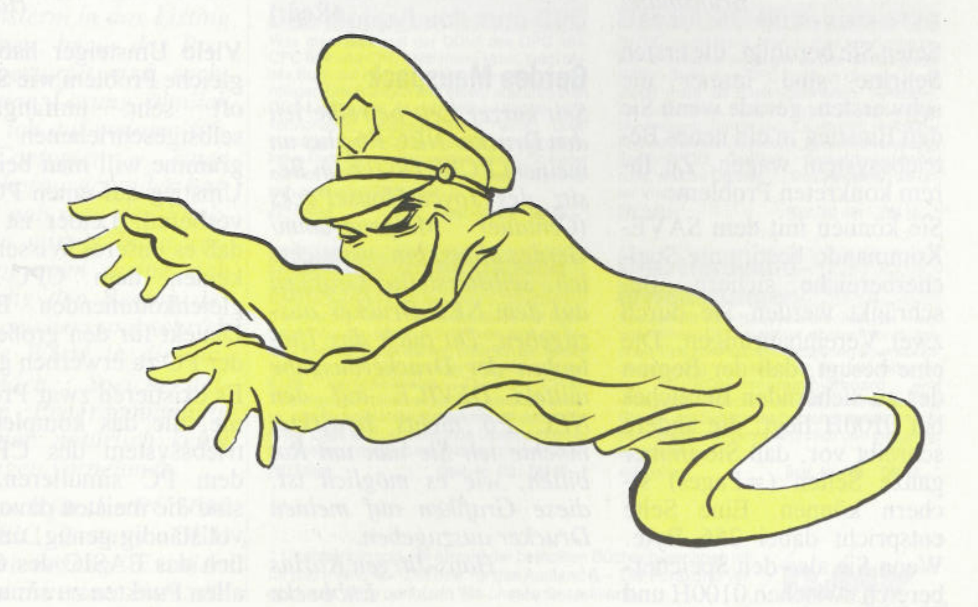
Eine kleine Mogelei führt dazu, daß man für den Kauf des kompletten Landes noch Geld bekommt. Das geht so: Nach dem Kauf einer Plantage wird wenig gekauft und nichts angebaut. Dann verkauft man das Land wieder, worauf der Bodenpreis ins Negative sinkt. Die Folge: Die kleinen Verluste, die man gemacht hat, gleichen sich durch den Kauf eines Landes zu negativen Preisen in nicht geahnten Maßen wieder aus.
Zu den Programmen, die aufgrund ihrer wenig durchdachten Programmierweise Spielraum für Mogeleien lassen, gehört sicher auch die Infocom-Westwood Produktion Mines of Titan.

Mines of Titan

Wer hier unter akuter Geldnot leidet, kann sich schnell und sicher mit Reichtümern beschenken. Dazu ist es angebracht, ein Casino aufzusuchen und vor dem Betreten den Spielstand zu sichern. Nun wirft man sich ins Vergnügen und entschließt sich, das saturnische Lotto zu spielen.
Das einzige, was hier zu tun ist, ist das Spiel in Gang zu bringen. Man sollte sich die Ausspielung, die nun folgt, notieren. Danach verläßt man die Spielhölle und lädt den vorher gesicherten Spielstand wieder ein. Das Casino wird erneut betreten, und beim Lotto wird der Höchsteinsatz auf die eben noch siegreichen sieben Zahlen gesetzt.
Sie werden ob Ihrer prophetischen Talente erstaunt sein, wenn genau diese Zahlen wie von Geisterhand aus dem Ziehungsgerät hervorkommen.
Mit der so gewonnenen Summe, sollte es keinerlei Probleme geben, das restliche Spiel ohne große Finanzsorgen zu überleben. Wenn es denn doch mal knapp werden sollte, dann geht es halt noch einmal in die Spielhalle. Das System ist wie bei keinem anderen Glücksspiel todsicher!

(Antje Hink/jf)





An unsere Leser

Die Rubrik »Leserbriefe« ist eine Einrichtung für alle Leser, die in irgendeiner Form Fragen, Probleme oder Anregungen zu Produkten, Programmierproblemen oder zu unserer Zeitschrift haben. Selbstverständlich sind wir bemüht, alle Leserfragen zu beantworten. Doch haben Sie bitte Verständnis, daß wir nicht alle eingehenden Briefe persönlich beantworten können. Oft erreichen uns mehrere Briefe zum gleichen Thema, einer davon wird dann stellvertretend für alle in unserer Zeitschrift beantwortet.

Ihre PC-Redaktion

Neue Durchwahlnummern für die Hotline

Für eilige Anfragen können Sie jetzt Ihren Redakteur direkt erreichen. Jeden Mittwoch von 17.00 – 20.00 Uhr stehen Ihnen zur Verfügung:

Claus Daschner (CPC)
☎ (0 56 51) 80 9 – 7 50

Joachim Freiburg (CPC/PC)
☎ (0 56 51) 8 09 – 7 51

Ralf Schößler (PCW)
☎ (0 56 51) 8 09 – 7 52

PS: Die Redaktion behält sich vor, Leserzuschriften in gekürzter Form wiederzugeben

Probleme mit CP/M

Zwar habe ich meinen CPC schon seit geraumer Zeit, doch erst vor kurzem habe ich Interesse an der Arbeit unter CP/M gefunden. Vorher habe ich die Systemdisketten eigentlich immer sehr stiefmütterlich behandelt (Formatieren oder Kopieren). Jetzt tauchen aber bereits die ersten Probleme mit dieser Benutzeroberfläche auf. Neben vielen generellen Fragen habe ich eine besonders dringende: Kann ich bestimmte Speicherbereiche auf einfache Weise auf Diskette bekommen? Wenn ja, wie mache ich das?

Bernhard Schnudersheim, Brunsbüttel

Seien Sie beruhigt, die ersten Schritte sind immer die schwersten, gerade wenn Sie den Einstieg in ein neues Betriebssystem wagen. Zu Ihrem konkreten Problem:

Sie können mit dem SAVE-Kommando bestimmte Speicherbereiche sichern. Beschränkt werden Sie durch zwei Vereinbahrungen. Die eine besagt, daß der Beginn des zu sichernden Bereiches bei 0100H liegt, die andere schreibt vor, daß Sie immer ganze Seiten (=Pages) sichern können. Eine Seite entspricht dabei 256 Byte. Wenn Sie also den Speicherbereich zwischen 0100H und 0FFFH, also exakt 4 KByte, abspeichern wollen müßten Sie dem SAVE-Kommando befehlen, 16 Seiten (16 * 256 Byte = 4 KByte) ab der Adresse 0100H zu Diskette zu bringen. Sie können das durch den Befehl

SAVE 16 Speicher.dat

erreichen. "Speicher.dat" ist dabei der Dateiname, unter dem die Daten gesichert werden.

Wenn Sie weitere interessante Tips und Tricks zum Thema CP/m interessieren, dann empfehlen wir Ihnen unser neues CPC Sonderheft 8'89/90. Hier werden viele Geheimnisse aus den Tiefen dieses Betriebssystems entschlüsselt.

(Red.)

Gerdes Mauspack

Seit kurzer Zeit betreibe ich den Drucker NEC P6 plus an meinem PCW. Da ich im Besitz des Joyce-MousePacks (Centauer Malprogramm/ Gerdes Maus) bin, versuchte ich, selbsterstellte Grafiken auf dem NEC-Drucker auszugeben. Da auch das Umlenken der Druckerausgabe mittels DEVICE auf den NEC P6 nichts bewirkte, möchte ich Sie nun um Rat bitten, wie es möglich ist, diese Grafiken auf meinen Drucker auszugeben.

Hans-Jürgen Karius Lübbecke

Da uns im Moment keine Möglichkeit bekannt ist, möchten wir Sie auf einen Artikel bezüglich der Programmierung von Hardcopyroutinen verweisen, welcher in einer der nächsten Ausgaben erscheinen wird. Obwohl dieser Artikel speziell für den CPC geschrieben ist, dürfte es für Sie keine Schwierigkeit darstellen die gewonnenen Informationen mittels der Basic-Erweiterung von Gerdes auf dieses umzusetzen.

Red.

Laufwerkeln (PCI 10/89)

Leider hat sich ein kleiner Fehler in dem sonst tollen Artikel Laufwerkeln eingeschlichen. Das Abspeichern des geänderten Betriebssystems kann in der Form nicht vonstatten gehen, da dem PCW die Schreibanweisung 'w' direkt hinter dem Doppelkreuz fehlte. Richtig muß die Schreibanweisung nun '#w **a:j14gcpm3.ems,0100,A0 FF'** *heißen.*

Rainer Tönnemann Oldenburg

CPC-BASIC auf dem PC?

Bitte teilen Sie mir mit, ob es das im Amstrad CPC installierte BASIC auch auf Diskette für den PC 1512 gibt und wo man gegebenenfalls solch eine Diskette zu welchem Preis erwerben könnte.

Georg Schenkel Hamburg

Viele Umsteiger haben das gleiche Problem wie Sie. Die oft sehr umfangreichen selbstgeschriebenen Programme will man bei einem Umstieg auf einen PC nicht verlieren. Leider ist es so, daß es unseres Wissens nach keinen dem CPC-BASIC gleichkommenden BASIC-Dialekt für den großen Bruder PC zu erwerben gibt.

Es existieren zwar Programme, die das komplette Betriebssystem des CPC auf dem PC simulieren, doch sind die meisten davon nicht vollständig genug, um wirklich das BASIC des CPC in allen Punkten zu simulieren. Klappt es doch, dann muß schon ein äußerst leistungsfähiger (und vor allem schneller) Prozessor her, um annähernd die Geschwindigkeit des CPC auf dem PC zu erreichen (so seltsam sich das auch anhören mag).

Wirklich zu empfehlen ist bei einem Umstieg das Erlernen eines anderen BASIC-Dialektes, etwa GW-BASIC oder BASIC 2 oder gar die Einarbeitung in eine für PCs leistungsstärkere Programmiersprache wie zum Beispiel Pascal.

Sie werden sehen, daß sich diese Empfehlung schlimmer anhört, als es sich schließlich herausstellt, denn auf Ihren CPC brauchen Sie letztlich nicht zu verzichten, der macht sich nämlich neben einem PC ausgezeichnet!

Red.

Ins Detail gegangen...

Ich betreibe einen CPC 464 mit DDI-1 Floppy, einer dk'tronics 64K-Speichererweiterung und einem Seikosha GP-500CPC-Drucker. Wenn ich voll beschriebene Seiten mit dem Textverarbeitungsprogramm ConText ausdrucke, passiert es oft, daß der Drucker die letzten fünf oder sechs Zeilen nicht ausdruckt. Nun würde mich interessieren, wieso dieses passiert. Vielleicht weiß auch ein anderer Benutzer dieses Druckers Rat oder hatte mit dieser Hardwarekonstellation ähnliche Probleme, die er gelöst hat. Für Lösungsvorschläge wäre ich sehr dankbar.

Jan Weller Zellertal

Diesem Aufruf an Besitzer gleicher Hardwarekonstellationen ist nichts hinzuzufügen. Wir sind gespannt...

Red.

Hercules-Probleme

Ich habe einen Amstrad PC 1512 mit zwei Laufwerken und eine Hercules-Grafikkarte mit entsprechendem Monitor.

Weder GEM noch BASIC 2 arbeiten mit der Hercules-Karte zusammen. Wie kann

ich es bewerkstelligen, daß sowohl GEM als auch BASIC 2 problemlos mit der Hercules-Karte zusammen laufen?

Wolf-Ulrich Loewe
Süsel

Sie sprechen da ein Problem an, das viele Anwender mit dem Amstrad PC haben. Die leider nur dürftig im Handbuch behandelte Installation der GEM-Benutzeroberfläche bereitet vor allem Einsteigern erhebliche Schwierigkeiten. Auch eventuelle Uminstallationen auf neue Grafikkonstallationen sind nur schwerlich zu meistern, wenn nur das Handbuch als Informationsquelle zur Verfügung steht.
Eine komplette Antwort innerhalb der Leserbriefe zu geben, wird aufgrund akuten Platzmangels ein nicht zu bewältigender Akt. Wir haben uns des Themas allerdings redaktionell angenommen und bringen schon in der nächsten Ausgabe eine grundsätzliche Installationshilfe für alle diejenigen, die in diesem Punkt Probleme haben.

Red.

Schreibmaschine am PPC

Ich habe mir vor kurzem einen Amstrad PPC 512/640 zugelegt. Durch Zufall habe ich dann vor wenigen Tagen Ihre Zeitschrift entdeckt. Leider ist für meinen PPC wenig darin, warum? Kann ich analog alles anwenden, was für den PC 1512/1640 angeboten wird?
Leider habe ich wenig Kenntnisse von PCs. Zur Zeit meines Studiums haben wir mit Lochkarten gearbeitet, und die Rechner füllten ganze Räume. Nun habe ich ein Problem: Wie erreiche ich, daß meine Schreibmaschine Olympia Carrera als Drucker am PPC benutzt werden kann? Ich habe alles nach Vorschrift über die Interface-Box angeschlossen. Nichts geht! Dabei läuft mein Star-Drucker LC10 einwandfrei. Was habe ich falsch gemacht oder nicht bedacht?

Ernst Meeser
Schwanewede

In der Tat haben wir den PPC in der Vergangenheit etwas stiefmütterlich behandelt. Unser redaktionelles Konzept für die nächsten Ausgaben sieht allerdings in diesem Punkt einen Richtungswechsel vor, so daß auch die Anwender des PPC in den nächsten Ausgaben Konkretes zu ihrem Computertyp finden werden. Das soll allerdings nicht heißen, daß die Programme und Tips, die wir bis jetzt veröffentlicht haben, nicht für Ihren PPC geeignet sind. Natürlich können Sie einen Großteil dieser Software auch für Ihren PPC nutzen.

Zu Ihrem speziellen Problem können wir schwerlich eine Ferndiagnose stellen, da wir mit dieser speziellen Computer-Druckerkonstellation keine Erfahrungen haben. Wir hoffen allerdings, daß einer unserer Leser über eine solche Hardwarezusammenstellung verfügt und uns seine Erfahrungen diesbezüglich mitteilt.

Red.

Speichermanipulation unter BASIC 2

Seit geraumer Zeit beschäftigt mich ein Problem, mit dem ich mich nun Sie wende.

Ich bin zur Zeit dabei, ein Programm zu schreiben, mit dem ich Funktionsgraphen komfortabel auf den Bildschirm bringe. Da ich es nicht befriedigend finde, den Funktionsterm in das Listing einzutippen, bevor das Programm gestartet wird, suchte ich eine Lösung ähnlich der, die ich auf meinem früheren Computer einen C-64 — herausgefunden hatte. Dort gab man den Funktionsterm mittels Inout ein. Das Programm suchte sich selbständig den Beginn des Arbeitsspeichers und schrieb dann die Werte in die entsprechenden Speicherstellen. Am Programmanfang durfte man natürlich keine Änderungen vornehmen.

Gibt es auch für den PC 1640 mit BASIC 2 unter GEM solch eine Lösung? Existiert vielleicht Litertur über die

Speicherorganisation von BASIC 2 unter GEM?

Thomas Schachtschneider
Lüneburg

Leider müssen wir Sie hinsichtlich der direkten Speichernmanipulation unter BASIC 2 sehr enttäuschen. Eine direkte Schnittstelle zur Maschinensprache oder direkten Speichermanipulation gibt es nicht. Alle beispielsweise aus dem CPC-BASIC bekannten Befehle wie CALL, POKE, INP etc. wurden unter BASIC 2 nicht berücksichtigt. Das hat natürlich Vor- und Nachteile, wobei die Nachteile in diesem Fall gerade für Personen mit erweiterten Programmierkenntnissen, die die Tiefen des Computers mehr nutzen wollen, überwiegen.

Red.

Rüge

Ich besitze einen PC 1512 mit zwei Laufwerken. Mit der Zeit wurde mir die Arbeit mit den Laufwerken zu unkomfortabel, daß ich auszog, eine Festplatte zu erstehen. Ich war in drei Läden, und jeder Verkäufer sagte mir, daß so etwas nicht ginge, da der Amstrad PC ein zu schwaches Netzteil besitze. Kann ich nun den Einbau einer Festplatte in den 1512 vergessen?

Mike Pilsena
Grebenstein

Um ein für allemal Schluß mit diesen Gerüchten zu machen: Es gibt keine Probleme mit dem Einbau einer Festplatte in den PC 1512! Eine besondere Rüge geht daher an alle diejenigen, die immer noch diese Fehlinformation verbreiten. Wir verweisen auf den Artikel "Datenmillionär" in diesem Heft, der sich genauer mit diesem Thema befaßt.

Red.

V30-Prozessor am PC 1512

Nach dem Einbau des V30-Prozessors bootet mein Amstrad PC 1512 nicht mehr von der Festplatte. Sie meldet sich zwar, doch danach stürzt der Rechner ab. Vom A-Laufwerk kann ich weiterhin normal booten. Vorerst habe ich den 8086er Prozessor wieder eingesetzt. Woran kann es liegen, daß der V30 nicht korrekt arbeitet?

Jörn Erdmann
Berlin

Wir vermuten, daß es sich weniger um ein Problem den V30-Prozessor betreffend handelt, sondern daß Ihr Festplatten-Controller Herd der Unruhe in Ihrem Computer ist. Es scheint der Fall zu sein, daß es sich bei jenem Controller um einen RLL-Controller handelt. Zwei Fehlerquellen sind dabei möglich:
Denkbar ist auf der einen Seite, daß der Controller einige zeitabhängige Tests vollzieht, was sich wegen der höheren Geschwindigkeit des V30-Prozessors im Vergleich mit dem 8086er als nicht positiv erweist.
Eine andere Schwierigkeit könnte sich aufgrund des nahezu gleichen Befehlssatzes von V30 und des 286er-Prozessors ergeben. Da sich diese beiden Prozessortypen, sieht man vom Protected Mode ab, wie zwei Geschwister gleichen, könnte der Festplatten-Controller durch ein paar etwas schlunzig programmierte Tests zu der Ansicht gelangen, bei Ihrem PC handele es sich um einen AT, so daß mit einem Male Dinge vom 1512 erwartet werden, die er nun wirklich nicht zu lösen imstande ist.
In jedem Fall empfehlen wir Ihnen den testweisen Umtausch des Festplatten-Controllers. Wenn sich auch bei anderen Controllern kein erfreuliches Ergebnis einstellt, sollten Sie den Händler aufsuchen und ihn direkt auf dieses Problem ansprechen. Wenn alle Gegenmaßnahmen fehlschlagen, bleibt Ihnen nichts anderes übrig, als beim 8086er-Prozessor zu bleiben, in diesem Falle haben Sie es dann wohl mit einem echten Montagsgerät zu tun, denn in der Regel gibt es keine Unstimmigkeiten in der harmonischen Verbindung zwischen Amstrad 1512 und V30-Prozessor.

Red.

Vokabeltrainer (Heft 3/89)

Sie übersandten mir auf Bestellung die PC-Databox 3/89. Leider läuft das Programm VOKABELN.BAS nicht. Bitte teilen Sie uns mit, welcher Befehl eingegeben werden muß, damit das Programm nicht im Abschnitt "Label nicht gefunden" stoppt.

Klaus Behrens
Einbeck

Der in Heft 3/89 vorgestellte Vokabeltrainer läuft nur dann zur vollständigen Zufriedenheit, wenn vorher auf dem aktuellen Datenträger ein Unterverzeichnis namens DICTION angelegt wird. Wird dieses vergessen, so gibt das Programm kurz nach dem Start die Fehlermeldung "Pfad nicht gefunden" aus, da programmintern auf dieses Unterverzeichnis zugegriffen wird.
Der Fehler liegt also nicht im Programm, sondern in der Vorbereitung. Daß wir diese offensichtlich nicht deutlich genug erklärt haben, bitten wir hiermit zu entschuldigen.

Red.

CPC-Datenrekorder MC 3810

Vor längerer Zeit kaufte ich bei der Firma Data Media den CPC-Datenrekorder "MC 3810". Mit diesem Gerät war und bin ich eigentlich immer noch zufrieden, wenn es nicht ein Manko zu beklagen gäbe: Das Gerät zeichnet keine Signale auf. Dieses fiel mir jedoch erst relativ lange Zeit nach dem Kauf auf; als ich bei der Vertriebsfirma Data Media um Hilfe bitten wollte, erfuhr ich, daß die Firma leider das Zeitliche gesegnet hatte. Mir blieb also zunächst nur die Möglichkeit, den Fehler selbst zu suchen (A/W-Schieber, A/W-Knopf usw.). Dieses blieb jedoch sowohl bei dem Gerät 3810 sowie auch bei einem später neu erworbenen Gerät 3810 A (welches sich scheinbar nur durch seine äußere Farbgebung vom 3810 unterscheidet) ohne Erfolg: auch der 3810 A zeichnet nicht auf. Als letztes blieb also nur noch die Überprüfung der Tape-Buchse meines CPC 664, welche ergab, daß dieser alle Signalwege laut Handbuch korrekt durchläuft (Signale werden ohne Fehler geladen und auch gesendet - hörbare Töne beim Test mit Wiedergabeverstärker).
Nun meine Frage: Ist der Fehler vielleicht an der beigefügten Verbindungsleitung des 3810 A zu suchen oder kennen Sie eine andere Fehlerquelle?

Michael Jung
Dortmund

Ohne Ihnen nahetreten zu wollen, Sie sollten Ihre Kaufgewohnheiten dahingehend ändern, daß Sie die Funktionstüchtigkeit eines Gerätes in vollem Umfang direkt nach dem Kauf in der Garantiefrist, vor allem aber vor dem Abtreten der Herstellerfirma, durchtesten.
Ihr Problem dürfte allerdings im vorliegenden Fall eher am Kabel liegen. Wir vermuten, daß Sie eines der gängigen Überspielkabel benutzen. Sie benötigen allerdings ein spezielles Überspielkabel oder einen Lötkolben, mit dem Sie die Verbindungen, wie im Handbuch beschrieben, umlöten. Dann dürfte sowohl der 3810 als auch der farblich modifizierte 3810 A voll funktionstüchtig sein.

Red.

Joyprint

Aufgrund einer Leseranfrage zu dem Programm Joyprint, welches in der Ausgabe 9/89 erschienen ist, hier kurz eine Problemlösung, welche sicherlich von allgemeinem Interesse ist.
Um ein Wort, einen Satz oder auch Teile davon nicht direkt am linken Rand des Papiers zu drucken, kann man sich der BASIC-Funktionen TAB(n) oder SPC(n) zur Positionierung bedienen. Diese Funktionen werden vom BASIC-Interpreter in eine dem Wert n entsprechende Anzahl Leerstellen umgewandelt und an den Drucker (in diesem Fall an JOYPRINT) geschickt. Um nun den darauf folgenden Text

invers oder unterstrichen darzustellen, darf die Umschaltung auf das neue Ausgabe-Attribut erst zu Beginn des Text-Ausdrucks erfolgen. Dies bedeutet, die erforderliche Steuer-Sequenz (zum Beispiel CHR$(27)"i"CHR$(1) für Farb-invertierte Ausgabe) darf erst nach der TAB(n)-beziehungsweise SPC(n)-Funktion in ein Ausgabe-Kommando 'eingebaut' werden.

Um das Wort 'invers' zum Beispiel an der Position 30 farbinvertiert auszudrucken, kann die folgende LPRINT-Kommandozeile benutzt werden:

```
LPRINT TAB(30)CHR$(27)"i"CHR$
(1)"invers"CHR$(27)"i"CHR$
(0)
```

Grundsätzlich sollte man am Ende der Zeile die eventuell benutzten Ausgabe-Attribute wieder zurückschalten, um Konflikte mit der nächsten Ausgabezeile beziehungsweise mit Zeichenvergrößerungen zu vermeiden.

Norbert Finke
Erndtebrück

Turbo-Pascal

Nachdem ich vor kurzem das Turbo-Pascal-Programm erstanden habe, habe ich mich nach intensivem Studium des Buches daran gemacht und eine Adreßverwaltung geschrieben. Sicherlich nichts Originelles, aber zum Einsteigen recht gut, finde ich. Nun, nachdem ich alle Routinen wie Eingeben, Suchen und Löschen fertiggestellt hatte, wollte ich mich daranmachen, eine Lade- beziehungsweise Speicherprozedur zu erstellen. Hier fehlen mir aber Informationen, wie ich dies bewerkstelligen kann, genauer gesagt, mit welchen Befehlen ich arbeiten muß.

Des weiteren würde mich interessieren, wie ich die Adressen auf dem Drucker ausgeben kann.

Alexander Grieß
Eschwege

Hier nun eine Lösung, welche wohl am einfachsten nachvollziehbar ist.

Die Prozedur zum Laden könnte wie folgt aussehen:

```
PROCEDURE Laden;
BEGIN
ASSIGN(datei,'ADR.DAT');
RESET(datei);
READLN(datei,name);
...
READLN(datei,ort);
counter:=counter+1;
CLOSE(datei)
END;
```

Die Prozedur zum Speichern sieht dann entsprechend so aus:

```
PROCEDURE Speichern
BEGIN;
ASSIGN(datei,'ADR.DAT');
REWRITE(datei);
WRITELN(datei,name);
...
WRITELN(datei,ort);
CLOSE(datei);
END;
```

Hierbei muß jedoch beachtet werden, daß dies nur für eine Adresse gilt.

Des weiteren muß die Variable 'Datei' als TEXT definiert sein.

Um diese Adressen nun auf den Drucker ausgeben zu können, fügen Sie bei dem WRITELN-Befehl einfach als Augabeeinheit ein LST ein. Die Druckzeile sieht also wie folgt aus:

```
WRITELN(LST,name);
```

Red.

Tastenspiele (PCI 1/90)

Die Idee, Steuerzeichen praktisch als Fehlermeldungen des CCP an den Bildschirm zu schicken, ist wirklich gut. Beim Ausprobieren sind mir eine ganze Reihe weiterer Möglichkeiten eingefallen. Wieso sollte man nicht die Steuerzeichen mit Hilfe des in CP/M vorhandenen TYPE-Befehls an den Bildschirm schicken?

Befinden sich zum Beispiel in der Datei 'CLS' die Zeichen **(↑Ä)H(↑Ä)E,** *so wird mit dem CP/M-Befehl TYPE CLS <RETURN> der Bildschirm gelöscht. Es ist praktisch, sich auf der RAM-Disk einige solcher 'Steuerdateien' zu halten. Neben CLS könnten da zum Beispiel Dateien wie 'INVON', 'INVOFF', 'WINDOW' und ähnliches stehen. Bei dieser Methode entfällt auch die Einschränkung, daß Großbuchstaben erzeugt werden. (INVON: (↑Ä)p). So können in Abbildung 1 auf Seite 64 in Heft 1/90 auch alle Werte zwischen 63 und 89 übergeben werden, für 75 wäre zum Beispiel 'k' zu setzen (siehe Handbuch 1 Anhang I/4). Auch das doch etwas lästige '?' zu Beginn jeder Operation entfällt, weil CP/M den Befehl TYPE ja ordnungsgemäß ausführen kann.*

Wolfgang Tremmel
Wiegenfeld

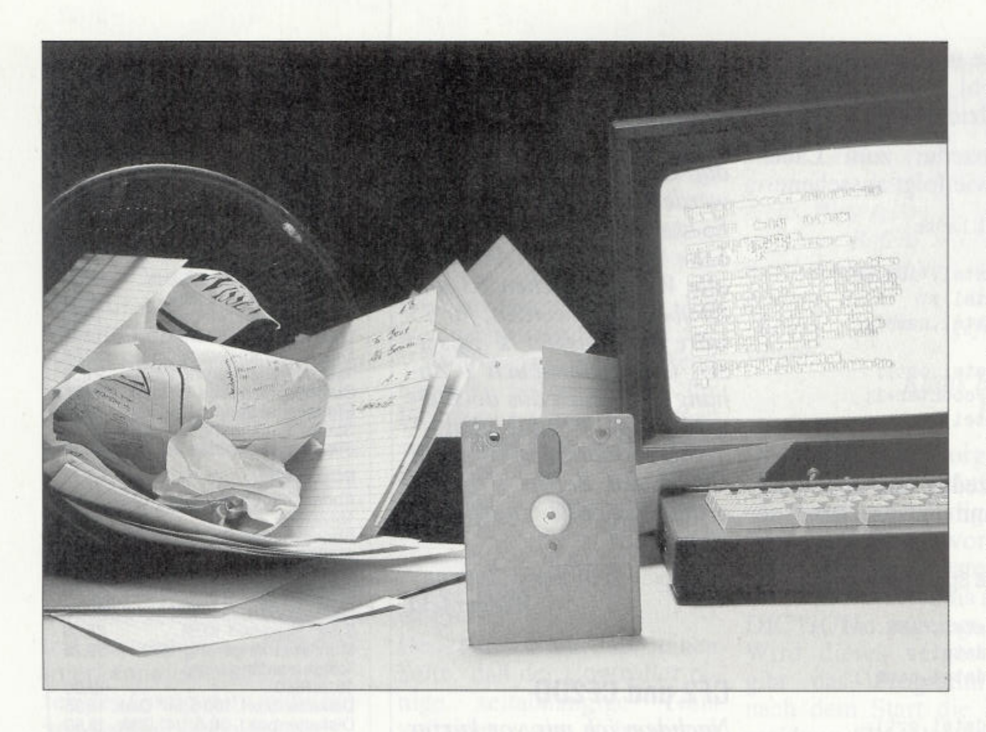
CF2 und CF2DD

Nachdem ich mir vor kurzer Zeit einen PCW zugelegt habe, möchte ich nun bezüglich der verschiedenen Disketten eine Frage an Sie richten: Ist es richtig, daß im Laufwerk B des PCW 8512 nur mit CF2DD-Disketten gearbeitet werden kann? Ein Test mit 'normalen' CF2-Disketten bei mir ergab, daß sich diese ohne Probleme im Zweitlaufwerk formatieren und beschreiben/lesen lassen.

Peter Steiner
Offenbach

Natürlich ist es möglich mit CF2-Disketten im B-Laufwerk zu arbeiten. Da diese jedoch nicht für ein so großes Diskettenformat konzipiert wurden, ist es möglich, daß des öfteren Schreib-/Lesefehler auftreten können.
Dies muß nicht, kann aber sein.
Deshalb empfiehlt es sich, so oft wie möglich Sicherheitskopien von diesen Disketten zu erstellen, um später keine bösen Überraschungen zu erleben. In der Redaktion, wo fast ausschließlich mit CF2-Disketten gearbeitet wird, ergaben sich bisher außer ein paar kleineren Diskettenfehlern keine größeren Schwierigkeiten, was sicherlich beweist, daß man nicht unbedingt CF2DD-Disketten benötigt.

(Red.)



dBase verständlich

Einsteigerkurs für das bekannte Datenbanksystem

In der letzten Ausgabe hatten wir uns mit dem Programmieren von kleinen dBase-Dateien und den Möglichkeiten des Sortierens von Daten befaßt.

dBase bietet aber eine ganz andere Möglichkeit des Sortierens. Es geht hier nicht ums Sortieren, sondern um das "Indizieren".

Das ist eine Technik, die speziell für Datenbanken entwickelt wurde. Auch hier wird nach einem "Schlüsselfeld" indiziert. Die Datei selbst bleibt aber in ihrer physikalischen Satzfolge unverändert. Dafür wird eine "Indexdatei" erstellt.

Diese besteht aus zwei Feldern. Das erste enthält die Werte des Schlüsselfeldes – jetzt aber in sortierter Reihenfolge – das zweite die zugehörigen Satznummern aus der eigentlichen Datei. Diese bleibt also unverändert erhalten. Bei einer sequentiellen Verarbeitung bestimmt aber jetzt die **Indexdatei** die Reihenfolge.

Für eine Datei können mehrere Indexdateien (über verschiedene Schlüsselfelder) erstellt werden. Als Schlüsselfelder werden meist Felder vom Typ "Character" gewählt. (Es können aber auch numerische Felder verwendet werden.)

Das Kommando zur Erzeugung einer Indexdatei lautet:

INDEX ON <schlüsselfeld> TO <indexdatei>

Die Bezeichnung der Indexdatei wählt man zweckmäßig so, daß das verwendete Schlüsselfeld zu erkennen ist. Von dBase erhält diese Datei die Typ-Erweiterung ".NDX".

Wir wollen jetzt unsere Kunden-Datei indizieren und zwar nach den Namen und nach Kundennummern. Wir geben ein:

USE KUNDEN INDEX ON NAME TO KUNAME

und dann

INDEX ON KUNDNR TO KUNUM.

Die erste Indexdatei hat also die Bezeichnung KUNAME.NDX, die zweite KUNUM.NDX. Das System meldet uns die Erstellung beider Indexdateien. Wenn jetzt die Datei KUNDEN mit einer Indexdatei aufgerufen wird, so gibt diese "den Ton an". Rufen Sie auf:

USE KUNDEN INDEX KUNAME

und später

USE KUNDEN INDEX KUNUM.

Überzeugen Sie sich mit dem Kommando LIST von der neuen Ordnung, obwohl die ursprünglichen Satznummern der Datei KUNDEN geblieben sind.

Wenn Sie jetzt die Kommandos GO TOP und GO BOTTOM anwenden und sich die eingestellten Sätze mit DISP ansehen, erkennen Sie, daß die "neue Ordnung" auch hier gilt.

Sie können auch beide Indexdateien gleichzeitig aufrufen:

USE KUNDEN INDEX KUNAME, KUNUM

Dann sind beide "im Spiel" und die erstgenannte ist **aktiv**. (Probe mit LIST.) Die zweite können Sie aktivieren mit SET INDEX TO KUNUM.

Sie können bis zu sieben Indexdateien für eine Datendatei erzeugen, aber nur eine davon kann aktiv sein.

Im Gegensatz zum Sortieren dauert die Erstellung einer Indexdatei auch bei sehr großen Dateien nur wenige Augenblicke.

Besonders angenehm wird es, wenn Sie neue Sätze hinzufügen oder Änderungen an Schlüsselfeldern der vorhandenen Sätze ausführen: Alle Indexdateien, die mit der Datei zusammen aufgerufen sind, werden automatisch mit geändert.

Das ist auch der Grund, weshalb Sie immer alle Indexdateien aufrufen sollten, auch wenn nur eine davon aktiv sein kann.

Sie können sich jetzt von diesen Eigenschaften der indizierten Dateien überzeugen, indem Sie zunächst die Datei KUNDEN mit ihren beiden Indexdateien aufrufen.

Fügen Sie dann mit dem Kommando APPEND neue Kunden hinzu. (Achten Sie dabei darauf, daß diese auch neue Kundennummern erhalten.)

Sie können auch mit dem Kommando EDIT in vorhandenen Sätzen den Namen (Schlüsselfeld) ändern. (Vielleicht hat eine Kundin geheiratet!)

Sie müssen dann wieder die betreffende Satznummer eingeben, die Sie aus einer Auflistung mit LIST entnehmen können.

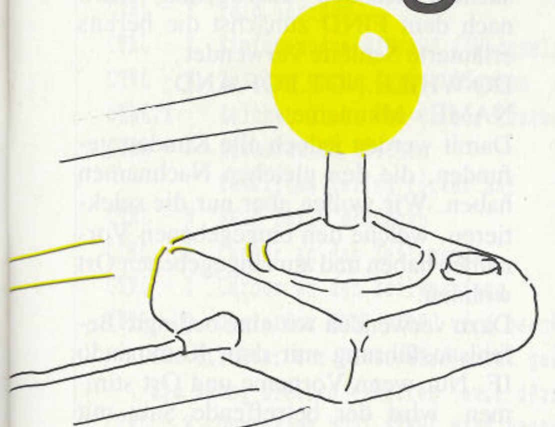
Editieren läßt sich auch in einer indizierten Datei mit dem Kommando "BROWSE"

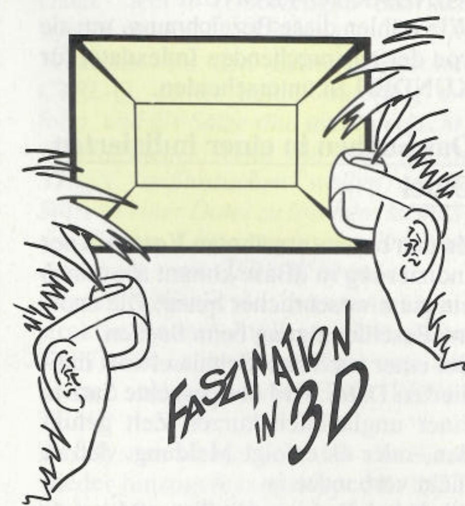
Sie sehen auch hier, daß in der Datei die durch den Index aktivierte Ordnung gilt.

Was tut man, wenn man doch einmal versehentlich eine Änderung in einer Datei durchgeführt hat, ohne die Indexdateien aufzurufen?

Die "Reparatur" ist ganz einfach: Nach dem erneuten Aufruf der Datei mit allen Indexdateien wird das Kommando REINDEX eingegeben.

CPC-SPIELE zu unglaublichen Preisen





3D-Light Cycle:

Das allseits bekannte und beliebte TRON-Spiel für zwei Personen jetzt wie im Film!! Erleben Sie dieses spannende und abwechslungsreiche Actionspiel jetzt in einer neuen Dimension. Verblüffende und noch nie dagewesene 3D-Effekte sowie professioneller Sound lassen Sie in eine völlig andere CPC-Welt versinken. Diesen Super-Hit muß man 'live' gesehen haben.

3D-Labyrinth:

Das beste Labyrinthspiel jetzt in Super-3D-Qualität. Auf der Suche nach Hinweisen zum Passwort, das den mächtigen Zentralcomputer lahmlegen kann, lauern tausend Gefahren auf Sie. Die räumliche, perfekte 3D-Darstellung, superschneller Grafikaufbau und viele Überraschungen garantieren eine völlig neue Art von Spielvergnügen.

Voraussetzungen:
CPC 464/664/6128 mit Farbmonitor. Das Spiel 3D-Light Cycle kann ohne 3D-Brille auch auf Grünmonitor gespielt werden, Darstellung dann in 2D.

INKLUSIVE 3D-Brille

3-Zoll-Diskette nur **39,– DM***

STARTEST

Action-Adventure für alle CPCs

Kassette jetzt nur 10,– DM*
Diskette jetzt nur 19,– DM*

Fantastic Four

Vier Superprogramme zum kleinen Preis:

- COCKAIGNE – Weltraum-Actionsspiel
- TERRANAUT I – Science-Fiction-Adventure
- FRUITS – Geschicklichkeitsspiel
- TERRANAUT II – Textadventure

Diskette jetzt nur 29,– DM*

CYRUS II Schach

Das bewährte Schachprogramm mit 3D-Display

Kassette jetzt nur 10,– DM*

SPECIAL OFFERS III

9 Spiele für alle CPCs

3 Disketten jetzt nur 39,– DM*

GAME BOX 3

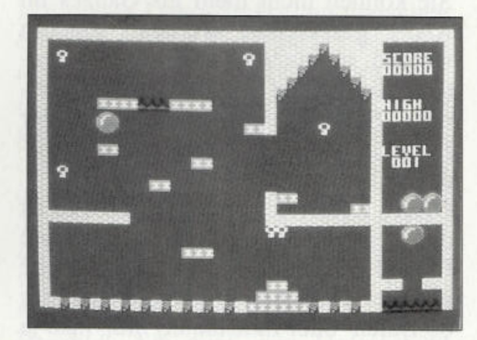
Vier Super-Programme zum kleinen Preis.

Die neue CPC-Spielebox enthält vier ausgesuchte TOP-Programme der Spitzenklasse. Da ist für jeden das Richtige dabei!

1. Alphajet
Lieben Sie Abenteuer und Gefahren im Weltraum? Dann steigen Sie in den neuentwickelten Alphajet und erforschen die interstellaren Sonnensysteme. Erleben Sie mit Alphajet ein hochklassiges Actionspiel mit außergewöhnlicher Grafik und vielen tollen Effekten.

2. Telefomania
Man schreibt das Jahr 1992. Das sogenannte Datenzeitalter hat begonnen. Aber einige Gebiete wurden bei der Vernetzung schlichtweg vergessen. Verhandeln Sie also direkt mit den Hausbesitzern über Tarife und Anschlußmöglichkeiten. Mit Telefomania erwartet Sie eine völlig neue Spielidee, die auch Sie begeistern wird.

3. Kampf den Insekten
Übernehmen Sie die Rolle eines berühmten Gärtners. Erschwert wird dies durch verschiedenartige Insekten, die immer wieder an Ihren herrlichen Pflanzen nagen und diese zerstören. Ein erlebnisreiches Geschicklichkeitsspiel für die ganze Familie mit vielen Überraschungen erwartet Sie.



4. Funbouncer
Ihr bester Freund wurde vom bösen Zauberer entführt. Als Sie das Schloß des Zauberers erreichen, werden Sie und Ihr Freund in einen feuerroten Ball verwandelt. In dieser Gestalt müssen Sie nun den geheimen Ausgang des Gewölbes finden, um den Fluch zu verlieren. Funbouncer ist ein schnelles und farbenfrohes Actionspiel mit vielen Levels und eigenem Bild-Construction-Set.

Für alle CPCs als Diskette 29,– DM*

* Unabhängig von der Anzahl der bestellten Produkte berechnen wir für das Inland DM 4,– bzw. für das Ausland DM 6,– Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege



Dann meldet das System nach kurzer Zeit, daß alles wieder ok ist.
Jetzt werden wir noch unsere Datei VERSGES nach den Kundennummern indizieren. (Das ist hier ein Sekundärschlüssel!)
USE VERSGES INDEX ON KUNDNR TO VKUNUM
Wir wählen diese Bezeichnung, um sie von der entsprechenden Indexdatei für KUNDEN zu unterscheiden.

Das Suchen in einer indizierten Datei

Zu den bereits erwähnten Vorteilen der Indizierung in dBase kommt aber noch ein ganz wesentlicher hinzu: Die enorme Beschleunigung beim Suchen.
Bei einer nach dem Schlüsselwort indizierten Datei wird der gesuchte Satz in einer unglaublich kurzen Zeit gefunden, oder es erfolgt Meldung, daß er nicht vorhanden ist.
Auch bei Dateien mit Tausenden von Sätzen dauert das nur wenige Sekunden. Wie kann man das erreichen?
Als Erklärung wird oft das "binäre Suchen" angegeben. Das ist aber nicht richtig.
Gesucht werden kann in einer Datei nur, wenn sie im RAM steht. Ob das dann sequentiell oder binär geschieht, ist ziemlich egal. Der Zugriff auf den RAM-Speicher ist so schnell, daß man den Unterschied praktisch nicht merkt.
Dateien mit Tausenden von Sätzen umfassen aber Hunderte von Kilobytes – ebenso die zugehörigen Indexdateien. Sie können nicht mehr als Ganzes im RAM gespeichert werden, sondern werden in einzelne "Blöcke" aufgeteilt, von denen jeweils einer von einem Externspeicher (Diskette oder Festplatte) in das RAM geladen werden muß.
Dieses Laden der einzelnen Blöcke ist der eigentliche zeitbeanspruchende Prozeß. Bei einer sequentiellen Suche in einer sortierten Datei (gleichgültig ob Daten- oder Indexdatei) muß man ja im Mittel immer die Hälfte aller Blöcke in den Speicher laden.
Man kann aber die **Indexdateien** (und nur diese) so organisieren, daß zum Suchen nur noch eine minimale Anzahl von Blöcken geladen werden muß. Man spricht dann von einer "Baumstruktur" (sogenannte "B-Bäume").
Nach der Theorie über das Suchen nun die Praxis. dBase hat dafür das Kommando
FIND <schlüsselwort>
Machen wir einige Versuche mit unserer Kundendatei:
USE KUNDEN INDEX KUNAME, KUNUM
aktiviert die nach Kundennamen indizierte Datei.
Jetzt geben wir ein: FIND "Homann". Das System meldet sich mit dem Prompt. Das ist hier das Zeichen, daß der gesuchte Satz "im Zugriff" ist. Mit DISPLAY können Sie sich davon überzeugen.
Sie können sich auch direkt ausgeben lassen, auf welche Satznummer der interne Satzzeiger des Systems zeigt. Benutzen Sie dazu das Ausgabe-Kommando "?", welches schon bei den Speichervariablen besprochen wurde
? # (dBase II) bzw.
? RECNO() (dBase III/III Plus)
Suchen Sie jetzt nach einem nicht vorhandenen Schlüsselwort (zum Beispiel FIND "Müller"). Das System meldet, daß die Suche erfolglos war. Der interne Satzzeiger steht auf Null. (# bzw. RECNO() = 0.)
Erfolglos bleibt die Suche natürlich auch, wenn das Schlüsselwort nicht so geschrieben ist, wie in der Datei.
FIND findet immer den ersten Satz mit dem gesuchten Schlüsselwort. Es können aber noch weitere folgen. Wie kann man sie finden, wenn man ihre Anzahl nicht weiß? Die beste Lösung bietet das Kommando DISPLAY mit der Option WHILE. Mit
DISPLAY WHILE <schlüsselfeld> = <schlüsselwort>
nach dem FIND erhält man genau alle weiteren Sätze.
Wir kennen WHILE von der DO WHILE-Schleife her. Es läuft so lange, wie die Sätze die Bedingung erfüllen. Bei der Indizierung funktioniert das bestens, weil hier alle entsprechenden Sätze aufeinander folgen.
Wenn wir die Suche programmieren, werden wir wieder eine DO WHILE-Schleife verwenden. Das Suchwort werden wir einer Speichervariablen zuweisen. Wir geben ein
STORE "Heinrichsen" TO Msuch
Wenn Sie jetzt das Kommando FIND mit Msuch verbinden wollen, so muß es heißen: FIND &Msuch
Sie müssen vor der Speichervariablen die sogenannte Makroersetzung "&" verwenden. Das ist eine Besonderheit des FIND-Kommandos, die man sich merken muß.
In dBase III/III Plus gibt es zusätzlich den Befehl SEEK, der sich "normal" verhält. Dort heißt es dann also
SEEK Msuch (oder auch FIND &Msuch)
Listing 1 zeigt Ihnen ein Programm zur Suche eines Kunden. Die Datei KUNDEN wird mit dem Index KUNAME aufgerufen. (KUNUM wird hier nicht benötigt.)
Name, Vorname und Anschrift werden mit ACCEPT eingelesen und in Suchvariablen gespeichert.
Um die Datensätze des (oder der) gesuchten Kunden auszugeben, wird nach dem FIND zunächst die bereits erläuterte Schleife verwendet
DO WHILE.NOT.EOF.AND. NAME=Mkuname
Damit werden jedoch alle Kunden gefunden, die den gleichen Nachnamen haben. Wir wollen aber nur die selektieren, welche den eingegebenen Vornamen haben und am eingegebenen Ort wohnen.
Dazu verwenden wir eine bedingte Befehlsausführung mit dem Kommando IF. Nur wenn Vorname und Ort stimmen, wird der betreffende Satz mit DISP OFF ausgegeben. Andernfalls wird die Ausgabe übersprungen.
Nach dem ENDIF (Ende der Bedingung) geht es in jedem Falle weiter. Jetzt muß der nächste Satz geprüft werden, ob er noch die Suchbedingung erfüllt. Dazu dient das Kommando SKIP, welches den nächsten Satz für die DO WHILE-Schleife bereitstellt.
(Etwas Ähnliches bewirkt ja in einer nicht indizierten Datei das Kommando CONTINUE in Verbindung mit LOCATE.)
Die Abfrage nach Vorname und Ort kann man übrigens **nicht** in die DO WHILE-Bedingung mit hineinnehmen. Diese Schleife wird ja beim ersten Satz, der die Bedingung nicht erfüllt, wieder verlassen!
Listing 2 zeigt Ihnen ein Programm zur Ausgabe eines "Vertragsspiegels". Für einen ausgewählten Kunden werden seine sämtlichen Verträge angezeigt.
Wir haben noch zwei Prüfungen eingebaut, die in der Praxis wichtig sind. Dafür verwenden wir wieder die Kommandos für eine bedingte Befehlsausführung, diesmal in der Form IF, ELSE, ENDIF.
Zunächst kann es vorkommen, daß der eingegebene Name nicht gefunden wird. Wir haben ja die Systemmeldungen ausgeschaltet, und das Programm würde dann ohne Kommentar "leerlaufen".
Wir haben oben gesehen, daß bei einem erfolglosen FIND der Satzzeiger auf Null steht. Das wird hier gleich nach dem FIND in einer IF-Bedingung abgefragt. Wenn ja, so wird eine Meldung ausgegeben, und das Programm ist schon beendet, weil es dann nach

CTRL - E	nach oben
CTRL - X	nach unten
CTRL - S	ein Zeichen zurück
CTRL - D	zum nächsten Zeichen
CTRL - A	ein Wort zurück
CTRL - F	zum nächsten Wort
CTRL - Y	Feld löschen
CTRL - V	Einfügemodus EIN/AUS ("Wechselschalter")
CTRL - G	Zeichen unter Cursor löschen
DELETE	Zeichen links vom Cursor löschen
CTRL - T	Cursorzeile löschen restliche Zeilen rücken auf
CTRL - N	Neue Zeile einfügen
CTRL - B	Cursor an das Zeilenende
CTRL - Z	Cursor an den Zeilenanfang
CTRL - Q	Operation MODI STRUC wird beendet die Struktur der Datei wird nicht geändert die Sätze bleiben erhalten (auch dBase II)
CTRL - W	Operation MODI STRUC wird beendet die Struktur der Datei wird geändert (dBase II: Die Sätze werden gelöscht)

Tabelle 1: Cursorsteuerung für Modify Structure Kommando

dem letzten ENDIF gleich zum Schluß kommt.

Wenn nein, dann folgt der Abschnitt nach dem ELSE, nämlich die Abfrageschleife

DO WHILE .NOT. EOF .AND. Mkuname$NAME

Da wir wieder nur die Kunden mit richtigem Vornamen und Anschrift haben wollen, kommt jetzt dieselbe IF-Abfrage wie in Listing 1. Ist es nicht der richtige Kunde, so geht es nach dem ENDIF Mvorname mit SKIP gleich zum nächsten Satz der Kundendatei.

Wenn der Kunde "stimmt", wird der Satz aus der Kundendatei ausgegeben, und für die Suche der zugehörigen Sätze in der Datei VERSGES wird die Kundennummer in einer Suchvariablen gespeichert.

Auch in der Datei VERSGES kann es vorkommen, daß keine Vertragssätze gefunden werden. Daher erfolgt hier eine gleichartige IF-Abfrage, wie bei der Datei KUNDEN.

Wenn Sie unsere Programme ausgiebig getestet haben, werden Sie einen "Schönheitsfehler" feststellen. Die Information wird auf dem Bildschirm "schmucklos" ausgegeben, so wie sie in der betreffenden Datei steht.

dBase hat die Möglichkeit, ansprechende Masken für die Bildschirmausgabe zu gestalten. Ebenso ist eine "formatierte" Druckerausgabe aus einer Datei möglich (auch aus mehreren Dateien einer Datenbank). In dieser einführenden Beitragsreihe können wir darauf leider nicht eingehen.

Weitere Kommandos

Es kommt vor, daß die Struktur einer Datei geändert werden muß, wenn ein Feld hinzugefügt oder die Länge eines vorhandenen geändert werden soll.

In dBase gibt es dafür den MODIFY STRUCTURE-Befehl. Bei seiner Anwendung in dBase II ist allerdings Vorsicht geboten! Dabei werden nämlich alle Sätze der Datei gelöscht. Um das zu vermeiden, müssen wir richtig vorgehen.

Als Beispiel wollen wir in der Datei KUNDEN das Feld VORNAME auf 15 Bytes erweitern. Wir rufen die Datei mit USE KUNDEN auf.

Jetzt müssen wir den Inhalt der Datei "sichern". Zu diesem Zweck speichern wir sie in einer "temporären" Datei mit dem Kommando

COPY TO TEMP ab.

Mit diesem Kommando wird die Datei KUNDEN vollständig (Struktur und Inhalt) in die neu erstellte Datei TEMP übertragen. (Rufen Sie TEMP auf und überzeugen Sie sich mit DISP STRUC und LIST.)

Jetzt können wir die Struktur von KUNDEN ändern. Sie müssen MODIFY STRUCTURE eingeben, und das System macht Sie auf die Datenlöschung aufmerksam und fragt noch einmal nach. Geben Sie "J" (oder "j") ein, so erscheint die Editiermaske für die Struktur. In Tabelle 1 sind die Tastencodes für die Cursorsteuerung angegeben.

Korrigieren Sie also die Länge des Feldes VORNAME. Wenn Sie jetzt den Editiermodus mit CTRL-W verlassen, ist die Änderung durchgeführt und die Datei "leer". (Überzeugen Sie sich wieder mit DISP STRUC.)

Verlassen Sie den Editiermodus mit CTRL-Q, so ist **keine** *Änderung erfolgt, und die Sätze sind* **nicht** *gelöscht. (Nachprüfen!) Wenn Sie also MODI STRUC "mißbrauchen" wollen, um die Sätze in einer Datei zu löschen, so müssen Sie den Editiermodus – ohne eine Änderung - mit CTRL-W beenden.*

Jetzt müssen wir die geänderte Dateistruktur wieder "füllen". Die Datei KUNDEN ist noch geöffnet und wir können eingeben: APPEND FROM TEMP

Das System meldet uns, daß alle Sätze wieder hinzugefügt wurden. Mit LIST können Sie sich überzeugen, daß die Felder wieder "gefüllt" sind.

Beachten Sie bitte, daß ein neues Feld, und auch ein Feld, dessen Namen Sie geändert haben, bei Anwendung von APPEND FROM leer bleibt.

Die Datei TEMP können Sie löschen mit

DELETE FILE TEMP

Sie darf dabei nicht geöffnet sein. Sie können Sie auch als "Backup-Datei" von KUNDEN aufheben. Dazu ändern Sie ihren Namen mit

RENAME TEMP TO KUNDEN.BAK ab.

Mit dieser Datei-Ergänzung erreichen Sie, daß die alte Kundendatei vom System nicht mehr "beachtet" wird.

In dBase III/III Plus brauchen Sie die geschilderten Schritte nicht selbst auszuführen. Sie werden automatisch vom System erledigt, wenn Sie MODI STRUC eingeben. Anschließend hat die Datei KUNDEN.DBF die neue und die vom System erstellte Datei KUNDEN.BAK die alte Struktur.

Beachten Sie auch den Unterschied zwischen COPY TO und APPEND FROM. Der eine Befehl ist nicht die Umkehrung des anderen!

COPY TO <zieldatei>

kopiert die geöffnete Datei in die Zieldatei. Wenn diese schon vorher existierte, ist ihre bisherige Struktur und auch der bisherige Inhalt "futsch".

APPEND FROM <quelldatei>

fügt die Sätze der Quelldatei zu den vorhandenen Sätzen der geöffneten Datei hinzu. Der bisherige Inhalt bleibt erhalten.

Die neuen Sätze kommen nur dann unverändert in der geöffneten Datei an, wenn die Quelldatei die gleiche Struktur hat. Nur wenn die geöffnete Datei leer ist, bildet APPEND FROM eine Art Umkehrung von COPY TO.
Eine Variante des Copy-Befehles ist COPY STRUCTURE TO <zieldatei>
Im Gegensatz zum "großen" Copy wird hier nur die Struktur übertragen. Dieser Befehl ist oft sehr nützlich, weil er Arbeit beim Erstellen einer neuen Datei erspart.
Hat man eine Datei, deren Struktur ungefähr "paßt", so kopiert man diese Struktur unter dem Namen der neuen Datei und ändert sie dann entsprechend ab.
Da die neue Datei noch keine Sätze enthält, kann man MODI STRUC ohne Bedenken anwenden.
Mit dem Kommando DISP FILES LIKE *.*
werden **alle** Dateien angezeigt, so wie beim DIR-Command in CP/M. Die anderen Möglichkeiten beim DIR-Command (z.B. *.CMD oder auch *.DBF) können Sie auch hier benutzen. Damit soll unser Streifzug durch das Datenbank-Verarbeitungssystem dBase beendet sein. Eine Vorstellung von seiner Leistungsfähigkeit werden Sie hoffentlich erhalten haben.

Literatur zu dBase

Wer sich auf einem 16-Bit-PC weiter mit dBase beschäftigen will, und dazu - nach unserem Vorschlag - die Version III Plus verwendet, wird (noch) keine Schwierigkeiten haben, sich entsprechende Literatur zu beschaffen.

Immerhin beginnt in den Buchhandlungen bereits die dBase-IV-Literatur derjenigen von III Plus den Rang abzulaufen.

Wer sich mit dBase II beschäftigen will (oder muß), ist sehr schlecht dran. Die Literatur dazu ist praktisch verschwunden.

Es gibt allerdings ein Buch, welches für **beide** Gruppen geeignet ist und sehr empfohlen werden kann (nicht nur wegen des günstigen Preises)

Peter Freese/Friedrich Müllmerstadt
Standardsoftware
Datenbanken
dBASE
Verlag rororo Computer

Peter Freese , Christian Kurtz
Software-Lexikon dBASE
Rororo-Verlag

J.Steiner
Schnellübersicht
dBase III Plus
Verlag Markt&Technik

Alexander Janson
dBase II, III und III Plus kompakt
Franzis Verlag

In diesem Buch findet man auch die dBase-II- und die Clipper-Kommandos.

Wer tiefer in die dBase-Programmierung einsteigen will, sollte sich unbedingt die "dBase Bibel" zulegen:

Luis Castro, Jay Hanson, Tom Rettig
Das dBase-Kompendium
(deutsche Übersetzung)
Verlag Markt&Technik

(Hans-Georg Sanner/cd)

```
******************** KSUCHEN3.CMD ******************************
********** Kundensuche Version 3, indizierte Kundendatei *********
*
*
SET TALK OFF
RELEASE ALL
USE KUNDEN INDEX KUNAME
*
   ERASE
   ?"Kunden aus Kunden-Datei suchen"
   ACCEPT "Name des Kunden eingeben " TO Mkuname
   ACCEPT "Vorname des Kunden eingeben " TO Mvorname
   ACCEPT "Anschrift des Kunden eingeben (Postleitzahl) " TO Mort
*
   FIND &Mkuname
     DO WHILE .NOT. EOF .AND. Mkuname$NAME
       IF Mvorname$VORNAME .AND. Mort$ORT
       DISPLAY OFF
       ENDIF
SKIP
    ENDDO WHILE .NOT. EOF
CLEAR
SET TALK ON
RETURN
```

```
****************** VERTSPGL.CMD ********************************
***** PROGRAMM ZUR ERSTELLUNG EINES VERTRAGS-SPIEGELS ************
*
SET TALK OFF
RELEASE ALL
USE KUNDEN INDEX KUNAME
SELECT SECONDARY
USE VERSGES INDEX VKUNUM
*
  SELECT PRIMARY
```

Listing dBase

```
   ERASE
   ?"Kunden aus Kunden-Datei suchen"
   ACCEPT "Name des Kunden eingeben " TO Mkuname
   ACCEPT "Vorname des Kunden eingeben " TO Mvorname
   ACCEPT "Anschrift des Kunden eingeben (Postleitzahl) " TO Mort
*
   FIND &Mkuname
   IF #=0
      ? "Kunde nicht gefunden. Name falsch! "
   ELSE
     DO WHILE .NOT. EOF .AND. Mkuname$NAME
       IF Mvorname$VORNAME .AND. Mort$ORT
        DISPLAY OFF
           STORE KUNDNR TO Mkunum
           ?"Verträge des Kunden aus Datei VERSGES suchen"
           SELECT SECONDARY
           ?
           FIND &Mkunum
           IF  #=0
              ? "kein Vertrag gefunden"
           ELSE
              DO WHILE .NOT. EOF .AND. Mkunum=KUNDNR
                 DISPLAY OFF
                 SKIP
              ENDDO WHILE .NOT. EOF
              ? "Datei-Ende"
           ENDIF
           SELECT PRIMARY
         ENDIF Mvorname
       SKIP
         ENDDO WHILE .NOT. EOF
     ENDIF
     CLEAR
     SET TALK ON
     RETURN
```

Listing dBase

AMS-Line

der direkte Draht zur Firma AMSTRAD

Wie jeden Monat präsentieren wir Ihnen die neuesten Informationen der Firma AMSTRAD.

Neuer Service-Partner

Da es trotz der bereits erfolgten "Bekanntmachungen" immer noch Kunden gibt, die sich mit Reparaturaufträgen an unseren alten Service-Partner wenden, möchten wir an dieser Stelle nochmals auf den erfolgten Wechsel hinweisen:
Wir haben uns von unserem alten Service-Partner getrennt. Die Firma Anders ist nicht mehr unser Service-Partner! Seit Anfang November – gleichzeitig mit der Einführung der 24monatigen Garantie auf unsere PCs – ist unser neuer Service-Partner die Firma
Telub Bitronic
Strahlenberger Weg 16
6000 Frankfurt 70
Tel. 069-6050140
Bitte wenden Sie sich dennoch nach wie vor mit Problemen zunächst an Ihren Fachhändler, der als Ihr direkter Ansprechpartner – vor allem auch für Garantieabwicklungen – zur Verfügung steht.

Preissenkung für Multimatic-Einzelblatteinzüge

Wie uns die Firma Wiedmann Unternehmensberatung mitteilt, können Sie Einzelblatteinzüge für die AMSTRAD-Drucker LQ 3500 und den Typenraddrucker des PCW 9512 jetzt zu sehr günstigen Preisen beziehen: den Einzug für den LQ 3500 zu einem Preis von 495,90 DM, den für den PCW- Drucker für 399,– DM. Mit Anfragen wenden Sie sich bitte an folgende Anschrift
Wiedmann Unternehmensberatung
Korbiniansplatz 2
8045 Ismaning
089-965029.

Patch für KEYBOARD.SYS – Punkt im Nummernblock

Unter den MS-DOS-Versionen 3.3 und 4.01 wird beim Laden des deutschen Tastaturtreibers im separaten Nummernblock ein "," eingestellt. Da es für einige Anwendungsprogramme aber notwendig ist, bei der Eingabe von Dezimalzahlen einen Punkt als "Dezimalkomma" einzugeben, gleichzeitig aber dennoch die Umlaute verfügbar sein sollen, erhielten wir bereits mehrfach Anfragen, ob hier andere Tastaturtreiber einsetzbar seien. Am einfachsten kommen Sie hier weiter, wenn Sie die Datei KEYBOARD.SYS patchen (natürlich nur mit Kopien arbeiten!).
Dafür kopieren Sie am besten die Datei KEYBOARD.SYS unter neuem Namen (z.B. KBGRPKT.SYS). Danach rufen Sie das Programm DEBUG auf:
DEBUG KBGRPKT.SYS
Der Eintrag für das Komma befindet sich in der Datei KEYBOARD.SYS der Version 3.3 an der Speicherstelle 1F64, bei der Version 4.01 an der Speicherstelle 282B. Um den Eintrag zu ändern, geben Sie folgendes ein (jeweils mit RETURN-Taste bestätigen):
e 282B
danach erscheint der aktuelle Eintrag
2C
direkt danach geben Sie den neuen Eintrag (ASCII-Code für Punkt) ein
2E
mit w wird die veränderte Datei abgespeichert, mit q verlassen Sie DEBUG.
Um den neuen Treiber zu laden, ändern Sie den Eintrag in der AUTOEXEC.BAT:
KEYB GR,,C:\DOS\KEYBOARD.SYS in
KEYB GR,,C:\DOS\KGBRPKT.SYS oder den entsprechenden Pfad und Dateinamen, den Sie für Ihren neuen Treiber verwendet haben.

Externes Laufwerke für PC1512 und PC1640

Sofern Sie ein Festplatten-Gerät besitzen und dennoch in den Genuß eines zweiten Diskettenlaufwerks kommen möchten, gibt es jetzt eine weitere "Fertiglösung". Die Firma Kosmalla & Partner bietet sowohl 3,5-Zoll-Laufwerke im Format 720 kByte als auch 5,25-Zoll-Laufwerke 360 kByte an. Sie kosten jeweils 398,– DM. Für diesen Preis erhalten Sie einen kompletten Einbausatz, bestehend aus Verteilerkabel für die Spannungsversorgung, Anschlußkabel mit Stecker auf Slotblech, Anschlußkabel für das externe Laufwerk und natürlich dem Laufwerk selbst. Die Einbauanleitung ist ausführlich gehalten und enthält auch die notwendigen Angaben zur Einbindung in der CONFIG.SYS-Datei.
Der Anbieter vertreibt direkt an Endkunden. Sie erhalten nähere Informationen unter der Anschrift
Kosmalla & Partner
Datenverarbeitung GmbH
Postfach 210806
6700 Ludwigshafen
0621-519749.

Externes Netzteil für PC 1512 und PC 1640

Da viele von Ihnen, zum Beispiel aufgrund von Erweiterung mit Grafikkarten, offensichtlich Probleme mit der Spannungsversorgung der Rechner haben, hier für die "Bastler" unter Ihnen noch einmal die technischen Daten des erforderlichen Netzteils und die Steckerbelegung. Manche Elektronikshops bieten zu sehr günstigen Preisen solche Netzteile an. Sie benötigen ein Schaltnetzteil mit mindestens 80 W Leistung, das Ihnen folgende Spannungen liefert: +5 V, -5 V, +12 V, -12V. Achten Sie bitte darauf, daß die angegebene Belegung des Netzteils auch korrekt ist. Sie könnten sonst "bleibende Schäden" an Ihrem Rechner verursachen! Das notwendige Anschlußkabel können Sie entweder über Ihren AMSTRAD-Fachhändler als Ersatzteil für den Monitor erhalten, oder Sie greifen auf die im Fachhandel erhältlichen Atari-Floppy-Stecker zurück. Die Belegung des Steckers ist folgende: Von der Lötseite des Steckers aus gesehen liegen die beiden Pins rechts oben neben der Kerbe (8 und 10) auf Masse, der nächste Pin (nach unten) ist frei, die beiden nächsten (4 und 2) liegen ebenfalls auf Masse, die beiden auf den Pin 1 (unten) folgenden (3 und 5) werden mit +5 V belegt. Diese drei "Paare" müssen aufgrund der hohen Strombelastung jeweils gebrückt werden! Links oben neben der Kerbe liegt Pin 11. Dieser wird mit +12 V belegt, der darunter folgende Pin 9 mit -12 V. Von den innenliegenden drei Pins ist nur der obere linke (Pin 13) mit -5 V belegt. Alle anderen Pins bleiben frei. Viel Erfolg beim Löten!
Wenn Sie keine genauen Angaben zum Netzteil erhalten können oder in Sachen "Hardware" nicht einige Vorkenntnisse haben, raten wir von einem Eigenbau ab. Wenden Sie sich in einem solchen Fall am besten an eine entsprechende Werkstatt!!!

Ihre

Kamela

Verwaltungshilfe

Rechnungen erstellen leichtgemacht

Gehören Sie auch zu den Leuten, die Tag für Tag Rechnungen mit der Hand schreiben müssen? Wenn ja, dann haben wir eine Hilfe für Sie. Mit diesem Programm können Sie auf leichte Weise Rechnungen erstellen und ausdrucken und so nebenbei Adressen verwalten.

Das Programm besteht aus zwei Teilen, das erste Listing ist das Installationsprogramm, wo alle wichtigen Daten festgelegt werden.

Dieses Programm braucht nur einmal gestartet zu werden, da die festgelegten Daten abgespeichert werden.

Nach Eingabe des Datums werden die drei Dateien SATZ.DAT, ADRESSEN.DAT und MWST.DAT geladen. Danach wird abgefragt, ob der Ausdruck <1> mit Briefkopf oder <2> ohne Briefkopf gewünscht wird. Bei Auswahl <1> mit Briefkopf wird vom Programm die Datei KOPF.DAT geladen. Anschließend erfolgt der Sprung zum Hauptmenü.

Das Hauptmenü ist wie folgt untergliedert:

Abrechnung, Stammdaten, Steuersatz, Kopfzeile, Ende

Hinweis: Wird eine der oben genannten Dateien nicht gefunden, erfolgt eine automatische Installation mit INSTALL.BAS.

Vom Hauptmenü können Sie mit Auswahl <2> das Menü STAMMDATEN anwählen.

ANLEGEN: Hier können Sie Ihre Kundenadressen einrichten und auf Diskette abspeichern.

Kundennummern werden vom Programm fortlaufend vorgegeben. Aus der Datei SATZ.DAT wird die Anzahl der Datensätze übernommen. Nach Eingabe <RETURN> in den Feldern 2 und 3 wird die Datenerfassung abgebrochen.

ANZEIGE: Seitenweise Anzeige der angelegten Datensätze. Mit der Cursortaste hoch ist jederzeit ein Abbruch oder Rückkehr zum Menü möglich.

DRUCKEN: Ausdruck Adressenliste.

ÄNDERN: Hiermit können Sie die einzelnen Adressen in den Feldern 2 bis 5 ändern. Die korrigierte Anschrift wird nach Eingabe <6> auf Diskette gespeichert.

HAUPTMENUE: Rückkehr Hauptmenü. Mit Auswahl <3> des Hauptmenüs werden die Mehrwertsteuersätze geändert und in die Datei MWST .DAT geschrieben. Menüpunkt <4> bietet Ihnen die Möglichkeit, Ihren individuellen Briefkopf, der beim Installieren angelegt wurde, zu ändern, und die Daten werden in die Datei KOPF.DAT geschrieben.

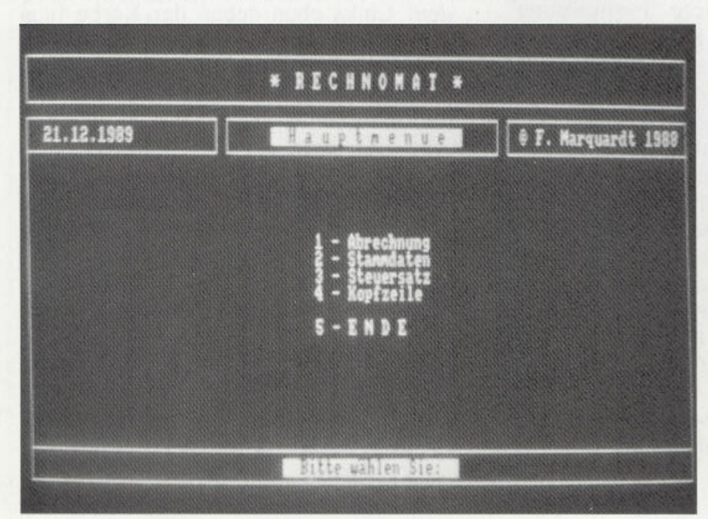
ABRECHNUNG: Zum eigentlichen Programmteil gelangen Sie mit Menüpunkt <1> des Hauptmenüs. Das Programm ist ausgelegt für 100 Einzelposten pro Rechnung/Gutschrift mit maximal zehn Textzeilen je Einzelposten.

1. Eingabe der Stammnummer des entsprechenden Kunden. Die Adresse wird auf dem Bildschirm angezeigt.
2. Auswahl: <R>ECHNUNG, <M>ENUE, <G>UTSCHRIFT, <S>TAMMNUMMER. Vom Programm wird <R>echnung vorgegeben und kann mit RETURN bestätigt werden. Bei Eingabe <M>enü erfolgt Rückkehr zum Hauptmenü. Eine neue Kundenadresse können Sie mit <S>tammnummer aufrufen.
3. Betreff eingeben – maximal sind vier Zeilen vorgesehen.
4. Textzeile eingeben. Durch Betätigen der Leertaste am Ende einer Textzeile erfolgt ein Sprung zur nächsten Zeile.
5. Menge eingeben.
6. Preis eingeben.
7. Auswahl: <S>umme ziehen oder nächste Eingabe <RETURN>.
8. Wird <S>umme gewählt, muß die Höhe des Mehrwertsteuersatzes bestimmt werden. (<1> volle Mwst, <2> ermäßigte Mwst, <3> ohne Mwst.)
9. Auswahl: <D>rucken oder zum <M>enü zurückkehren.
10. Bei Auswahl <D>rucken wird nach Anzahl der Ausdrucke gefragt. Vorgabe 4 kann mit RETURN bestätigt oder überschrieben werden. Der Ausdruck wird vom System so gesteuert, daß am Ende jeder Seite automatisch ein Übertrag gezogen wird.
11. Nach Ausdruck der Rechnung/ Gutschrift besteht die Möglichkeit, nochmals zu <D>rucken oder zum <M>enü zurückzukehren.
12. Prozentuale Auf- und Abschläge auf die **jeweils letzte Summe** (z.B. 10% Rabatt etc.) sind wie folgt möglich:

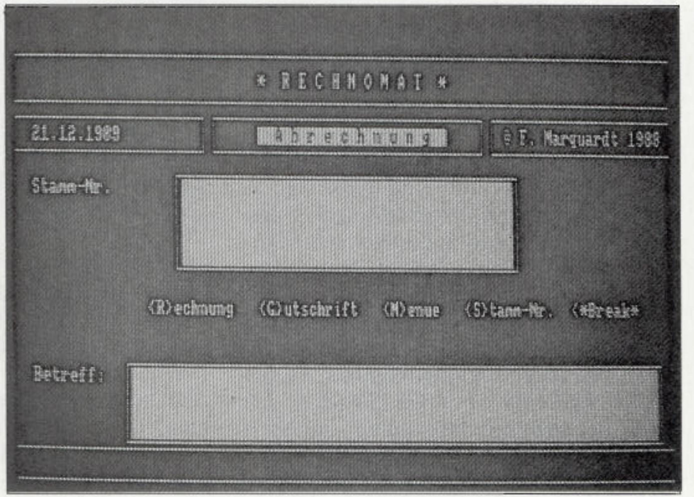
1. Textzeile(n) eingeben.
2. Eingabefelder für Menge und Preis mit <RETURN> überspringen.
3. Prozentsatz eingeben (Abzug als Minus -10).
4. Auswahl <S>umme ziehen oder nächste Eingabe RETURN.

Mit Auswahl <5> des Hauptmenüs wird das Programm beendet.

(Friedrich Marquardt/cd)



Das Hauptmenü von Rechnomat



Das Abrechnungsmenü: Nach Eingabe der Stammnummer wird die komplette Anschrift ausgegeben.

für 6128



```
10 REM ********************************** [1835]
***
20 REM *******  Installation Rechnomat *** [1860]
***
30 REM ********************************** [1835]
***
40 MODE 2                                 [513]
50 SYMBOL AFTER 90                        [1214]
60 SYMBOL 91,&66,&18,&3C,&66,&7E,&66,&66,& [1766]
0
70 SYMBOL 92,&66,&3C,&66,&66,&66,&66,&3C,0 [2126]
80 SYMBOL 93,&66,&44,&66,&66,&66,&66,&3C,0 [2286]
90 SYMBOL 123,&CC,0,&78,&C,&7C,&CC,&76,0  [2418]
100 SYMBOL 124, &66,0,&3C,&66,&66,&66,&3C, [2274]
0
110 SYMBOL 125,&66,0,&66,&66,&66,&66,&3E,0 [1843]
120 SYMBOL 126,&3C,&66,&66,&6C,&66,&66,&6C [2404]
,&60
130 KEY DEF 17,1,125,93                   [746]
140 KEY DEF 19,1,123,91                   [1140]
150 KEY DEF 22,1,124,92                   [1244]
160 KEY DEF 26,1,124,92,64                [765]
170 KEY DEF 24,1,126,94,0                 [1480]
180 WINDOW#4,3,78,7,22                    [1338]
190 MOVE 10,396:DRAWR 620,0:DRAWR 0,-40:DR [3726]
AWR -620,0:DRAWR 0,40
200 LOCATE 30,2:PRINT"*  R E C H N O M A T [2754]
 *"
210 MOVER 0,-54:DRAWR 170,0:DRAWR 0,-30:DR [3136]
AWR -170,0:DRAWR 0,30
220 MOVER 180,0:DRAWR 260,0:DRAWR 0,-30:DR [2714]
AWR -260,0:DRAWR 0,30
230 MOVER 270,0:DRAWR 170,0:DRAWR 0,-30:DR [4626]
AWR -170,0:DRAWR 0,30
240 LOCATE 4,5:PRINT D$:LOCATE 60,5:PRINT  [3863]
CHR$(164)" F. Marquardt 1988"
250 MOVE 10,310:DRAWR 0,-300:DRAWR 620,0:D [1992]
RAWR 0,300
260 MOVE 10,38:DRAWR 620,0                [1100]
270 LOCATE 26,5:PRINT CHR$(24)"       *  In [3929]
stallation  *        "CHR$(24)
280 LOCATE 28,15:PRINT"<E>nde       <RETURN> [2207]
 = weiter"
290 LOCATE 30,24:PRINT CHR$(24)"      Bitte [3732]
w{hlen Sie:   "CHR$(24)
300 PRINT CHR$(7):LOCATE 20,13:PRINT"Vorsi [6283]
cht  bestehende Dateien werden gel¦scht"
310 opt$="e"+CHR$(13)                     [1250]
320 GOSUB 880                             [1060]
330 ON K GOTO 340,350                     [1090]
340 LOCATE 28,15:PRINT STRING$(29,32):LOCA [6380]
TE 20,13:PRINT"          Installation abgeb
rochen          ":FOR i=1 TO 1000:NEXT:C
LS:END
350 LOCATE 20,13:PRINT STRING$(45,32)     [1376]
360 LOCATE 28,15:PRINT STRING$(29,32)     [1086]
370 LOCATE 30,24:PRINT CHR$(24)"   Datens{t [4483]
ze einrichten "CHR$(24)
380 LOCATE 30,11:PRINT"Anzahl Datens{tze"  [4120]
390 LOCATE 49,11:INPUT"",HH               [997]
400 DIM x(HH),stamm$(HH),Anr$(HH),nam$(HH) [2325]
,strass$(HH),ort$(HH)
410 LOCATE 26,11:PRINT"Datei   SATZ.DAT  wi [3228]
rd angelegt"
420 OPENOUT"SATZ.DAT"                     [1319]
430 WRITE#9,HH                            [915]
440 CLOSEOUT                              [902]
450 LOCATE 24,13:PRINT"Datei  ADRESSEN.DAT [3601]
 wird angelegt"
460 OPENOUT"ADRESSEN.DAT"                 [1617]
470 FOR x=1 TO HH                         [829]
480 WRITE#9,stamm$(x),Anr$(x),nam$(x),stra [3602]
ss$(x),ort$(x)
490 NEXT                                  [350]
500 CLOSEOUT                              [902]
510 LOCATE 26,11:PRINT STRING$(35,32)     [1732]
520 LOCATE 24,13:PRINT STRING$(35,32)     [1519]
530 LOCATE 30,24:PRINT CHR$(24)"  S t e u  [3101]
e r s a t z  "CHR$(24)
540 LOCATE 29,13:PRINT"Voller Steuersatz:  [3995]
    %"
550 LOCATE 29,14:PRINT"Erm{ss.Steuersatz:  [3325]
    %"
560 LOCATE 48,13:INPUT"",STEUER           [1479]
570 LOCATE 49,14:INPUT"",STEUER1          [2179]
580 OPENOUT"MWST.DAT"                     [915]
590 WRITE#9,STEUER,STEUER1                [2979]
600 CLOSEOUT                              [902]
610 LOCATE 30,24:PRINT CHR$(24)"   K o p f [4563]
 z e i l e  "CHR$(24)
620 CLS#4                                 [386]
630 LOCATE#4,1,5                          [360]
640 PRINT#4,TAB(10);"Firma 1. Zeile  "CHR$ [2756]
(246)TAB(52)CHR$(247)
650 PRINT#4,TAB(10);"        2. Zeile  "CHR$ [2780]
(246)TAB(52)CHR$(247)
660 PRINT#4,TAB(10);"Stra^e           "CHR$ [2295]
(246)TAB(52)CHR$(247)
670 PRINT#4,TAB(10);"PLZ, Ort         "CHR$ [4008]
(246)TAB(52)CHR$(247)
680 PRINT#4,TAB(10);"Telefon          "CHR$ [3262]
(246)TAB(52)CHR$(247)
690 PRINT#4,TAB(10);"Bankverbindung   "CHR$ [3696]
(246)TAB(52)CHR$(247)
700 PRINT#4,TAB(10);"Konto-Nr.        "CHR$ [2997]
(246)TAB(52)CHR$(247)
710 PRINT#4,TAB(10);"Bankleitzahl     "CHR$ [2688]
(246)TAB(52)CHR$(247)
720 LOCATE 30,11:LINE INPUT"",Vorname$    [1933]
730 LOCATE 30,12:LINE INPUT"",Name$       [1349]
740 LOCATE 30,13:LINE INPUT"",Strasse$    [1484]
750 LOCATE 30,14:LINE INPUT"",Plz$        [1265]
760 LOCATE 30,15:LINE INPUT"",Telefon$    [1326]
770 LOCATE 30,16:LINE INPUT"",Bank$       [2206]
780 LOCATE 30,17:LINE INPUT"",Kto$        [1684]
790 LOCATE 30,18:LINE INPUT"",Blz$        [1692]
800 CLS#4                                 [386]
810 LOCATE 26,10:PRINT"Datei  KOPF.DAT  wi [3140]
rd angelegt"
820 OPENOUT"KOPF.DAT"                     [1149]
830 WRITE#9,Vorname$,Name$,Strasse$,Plz$,T [2780]
elefon$,Bank$,Kto$,Blz$
840 CLOSEOUT                              [902]
850 LOCATE 30,24:PRINT CHR$(24)"  Installa [4841]
tion beendet  "CHR$(24)
860 LOCATE 26,10:PRINT"        RM.BAS  wird g [3547]
eladen     "
870 RUN"RM.BAS"                           [812]
880 K$=INKEY$:IF K$="" THEN 880           [1311]
890 IF INSTR(UPPER$(opt$),UPPER$(K$))=0 TH [1827]
EN 880
900 K=INSTR(UPPER$(opt$),UPPER$(K$))      [705]
910 RETURN                                [555]
```

Listing RECHNOMA

```
10 REM ********************************** [2263]
**********************
20 REM *****  Rechnomat CPC 6128 mit Stamm [2773]
datenverwaltung  *****
30 REM *****                Version 2.0   [1660]
                    *****
40 REM ********************************** [2263]
**********************
50 MODE 2                                 [513]
60 ON ERROR GOTO 5140                     [1526]
70 INPUT"Datum: ",D$                      [1398]
80 CLS:LOCATE 30,12:PRINT CHR$(24)"SATZ.DA [4054]
T"CHR$(24)" wird eingelesen"
90 OPENIN"SATZ.DAT"                       [682]
100 INPUT#9,HH                            [777]
110 CLOSEIN                               [752]
120 LOCATE 30,12:PRINT CHR$(24)"MWST.DAT"C [2525]
HR$(24)
130 OPENIN"MWST.DAT"                      [2063]
```

Listing RECHNOMA

```
140 INPUT#9,STEUER,STEUER1                        [1505]
150 CLOSEIN                                       [752]
160 DIM a$(100),a1$(100),a2$(100),a3$(100)        [6615]
,a4$(100),a5$(100),a6$(100),a7$(100),a8$(1
00),a9$(100),m(100),ap(100),ep(100),p(100)
,prozent(100)
170 DIM stamm$(HH),anr$(HH),nam$(HH),stras        [3079]
s$(HH),ort$(HH),Satz(HH)
180 LOCATE 26,12:PRINT CHR$(24)"ADRESSEN.D        [3307]
AT"CHR$(24)
190 OPENIN"ADRESSEN.DAT"                          [957]
200 FOR x=1 TO HH                                 [829]
210 INPUT#9,stamm$(x),Anr$(x),Nam$(x),stra        [2704]
ss$(x),ort$(x)
220 NEXT                                          [350]
230 CLOSEIN                                       [752]
240 SYMBOL AFTER 90                               [1214]
250 SYMBOL 91,&66,&18,&3C,&66,&7E,&66,&66,        [1766]
&0
260 SYMBOL 92,&66,&3C,&66,&66,&66,&66,&3C,        [2126]
0
270 SYMBOL 93,&66,&44,&66,&66,&66,&66,&3C,        [2286]
0
280 SYMBOL 123,&CC,0,&78,&C,&7C,&CC,&76,0         [2418]
290 SYMBOL 124, &66,0,&3C,&66,&66,&66,&3C,        [2274]
0
300 SYMBOL 125,&66,0,&66,&66,&66,&66,&3E,0        [1843]
310 SYMBOL 126,&3C,&66,&66,&6C,&66,&66,&6C        [2404]
,&60
320 KEY DEF 17,1,125,93                           [746]
330 KEY DEF 19,1,123,91                           [1140]
340 KEY DEF 22,1,124,92                           [1264]
350 KEY DEF 26,1,124,92,64                        [765]
360 KEY DEF 24,1,126,94,0                         [1480]
370 U$="#######,.##"                              [689]
380 WINDOW#7,15,77,19,22:PAPER#7,1:PEN#7,2        [3141]
390 WINDOW#1,4,24,10,16                           [1682]
400 WINDOW#6,1,80,23,25                           [970]
410 WINDOW#2,21,60,10,16:PAPER#2,1:PEN#2,2        [2166]
420 WINDOW#4,3,78,7,22                            [1338]
430 WINDOW#5,3,80,13,25                           [1391]
440 CLS:PRINT"Ausdruck:       <1> mit  Br         [6410]
iefKopf      <2> ohne Briefkopf"
450 opt$="12"                                     [83]
460 GOSUB 680                                     [972]
470 druck=K                                       [530]
480 ON K GOTO 5510,490                            [1758]
490 CLS                                           [91]
500 MOVE 10,396:DRAWR 620,0:DRAWR 0,-40:DR        [3726]
AWR -620,0:DRAWR 0,40
510 LOCATE 30,2:PRINT"*  R E C H N O M A T        [2754]
 *"
520 MOVER 0,-54:DRAWR 170,0:DRAWR 0,-30:DR        [3136]
AWR -170,0:DRAWR 0,30
530 MOVER 180,0:DRAWR 260,0:DRAWR 0,-30:DR        [2714]
AWR -260,0:DRAWR 0,30
540 MOVER 270,0:DRAWR 170,0:DRAWR 0,-30:DR        [4626]
AWR -170,0:DRAWR 0,30
550 LOCATE 4,5:PRINT D$:LOCATE 60,5:PRINT         [3863]
CHR$(164)" F. Marquardt 1988"
560 MOVE 10,310:DRAWR 0,-300:DRAWR 620,0:D        [1992]
RAWR 0,300
570 MOVE 10,38:DRAWR 620,0                        [1100]
580 LOCATE 30,5:PRINT CHR$(24)"  H a u p t        [4489]
 m e n u e  "CHR$(24)
590 LOCATE 35,11:PRINT"1 - Abrechnung"            [2698]
600 LOCATE 35,12:PRINT"2 - Stammdaten"            [1856]
610 LOCATE 29,13:PRINT"       3 - Steuersat       [2139]
z      "
620 LOCATE 29,14:PRINT"       4 - Kopfzeile"      [2956]
630 LOCATE 35,16:PRINT"5 - E N D E    "           [1932]
640 LOCATE 31,24:PRINT CHR$(24)"  Bitte w{        [2664]
hlen Sie:  "CHR$(24)
650 opt$="12345"                                  [381]
660 GOSUB 680                                     [972]
670 ON k GOTO 730,3920,5000,5180,720              [1942]
680 K$=INKEY$:IF K$="" THEN 680                   [1445]
690 IF INSTR(UPPER$(opt$),UPPER$(K$))=0 TH        [1739]
EN 680
700 K=INSTR(UPPER$(opt$),UPPER$(K$))              [705]
710 RETURN                                        [555]
720 CLS:LOCATE 25,13:PRINT"P r o g r a m m        [4518]
 e n d e":FOR i=1 TO 1000:NEXT:CLS:END
730 GP=0:GP1=0                                    [473]
740 LOCATE 31,24:PRINT STRING$(21,32)             [1127]
750 LOCATE 30,5:PRINT CHR$(24)"  A b r e c        [4158]
 h n u n g  "CHR$(24)
760 CLS#4                                         [386]
770 WINDOW#3,21,60,8,12                           [1411]
780 PAPER#3,1:PEN#3,2:CLS#3                       [1510]
790 MOVE 155,205:DRAWR 0,85:DRAWR 329,0:DR        [1859]
AWR 0,-85:DRAWR -329,0
800 LOCATE 4,8:PRINT"Stamm-Nr."                   [2622]
810 LOCATE #5,66,3:PRINT#5,"< >"                  [1771]
820 LOCATE 17,15:PRINT"<R>echnung    <G>uts       [5252]
chrift    <M>enue    <S>tamm-Nr."
830 LOCATE 4,19:PRINT"Betreff:"                   [2671]
840 CLS#7                                         [383]
850 MOVE 108,45:DRAWR 0,70:DRAWR 511,0:DRA        [2897]
WR 0,-70:DRAWR -511,0
860 LOCATE 14,8:INPUT"",such$                     [1212]
870 FOR x=1 TO HH                                 [829]
880 IF x=HH AND INSTR(stamm$(x),such$)=0 T        [2927]
HEN 960
890 IF INSTR(stamm$(x),such$)=0 THEN 950          [1671]
900 PRINT#3," ";ANR$(x)                           [867]
910 PRINT#3," ";Nam$(x)                           [819]
920 PRINT#3," ";strass$(x)                        [817]
930 PRINT#3:PRINT#3," ";Ort$(x)                   [2287]
940 GOTO 970                                      [330]
950 NEXT                                          [350]
960 LOCATE 31,24:PRINT CHR$(7)" Satz nicht        [8309]
 in Datei ":FOR i=1 TO 1000:NEXT:LOCATE 31
,24:PRINT STRING$(21,32):LOCATE 14,8:PRINT
"    "
970 LOCATE#5,67,3                                 [661]
980 PRINT#5,"R"                                   [787]
990 LOCATE#5,67,3                                 [661]
1000 INPUT#5,"",R$                                [794]
1010 anr$=anr$(x):nam$=nam$(x):strass$=str        [2581]
ass$(x):ort$=ort$(x)
1020 R$=UPPER$(R$)                                [2071]
1030 IF R$="R" OR R$="G" OR R$="" THEN 107        [1994]
0
1040 IF R$="S" THEN CLS#3:LOCATE 14,8:PRIN        [3283]
T"    ":GOTO 860
1050 IF R$="M" THEN CLS#4:GRAPHICS PEN 2:M        [6361]
OVE 108,47:DRAWR 0,-2:DRAWR 511,0:DRAWR 0,
2:GRAPHICS PEN 1:GOTO 580
1060 IF R$<>"S" OR R$<>"G" OR R$<>"M" OR R        [5742]
$<>"R" OR R$<>"" THEN PRINT CHR$(7):GOTO 9
70
1070 LINE INPUT#7," ",b$                          [1722]
1080 LINE INPUT#7," ",b1$                         [1719]
1090 LINE INPUT#7," ",b2$                         [1717]
1100 LINE INPUT#7," ",b3$                         [1731]
1110 CLS                                          [91]
1120 WINDOW#3,1,80,4,21                           [1150]
1130 PAPER#3,2:PEN#3,1                            [1644]
1140 IF R$ ="" OR R$="R" THEN R$ ="R e c h        [2227]
 n u n g"
1150 IF R$ ="G" THEN R$ ="G u t s c h r i         [2160]
 f t"
1160 PRINT CHR$(24)" "R$" "CHR$(24);              [1741]
1170 PRINT TAB(42)"%"TAB(48)"MENGE"TAB(59)        [3090]
"PREIS"TAB(71)"BETRAG DM"
1180 PRINT STRING$(79,154)                        [1337]
1190 LOCATE 1,22                                  [730]
1200 PRINT STRING$(79,154)                        [1337]
1210 FOR a=1 TO 100                               [909]
1220 CLS#6                                        [384]
1230 LOCATE 43,23:PRINT".      n{chste Textz      [2880]
eile  <LEERTASTE>"
1240 LINE INPUT#6,"1 TEXT: ",A$(a)                [2246]
1250 IF RIGHT$(A$(a),1)=CHR$(32) THEN 1260        [2071]
 ELSE 1430
1260 LINE INPUT#6,"2 TEXT: ",A1$(a)               [2685]
1270 IF RIGHT$(A1$(a),1)=CHR$(32) THEN 128        [2610]
0 ELSE 1430
1280 LINE INPUT#6,"3 TEXT: ",A2$(a)               [2377]
1290 IF RIGHT$(A2$(a),1)=CHR$(32) THEN 130        [2711]
0 ELSE 1430
1300 CLS#6:LOCATE 43,23:PRINT".      n{chste      [5454]
 Textzeile  <LEERTASTE>":LINE INPUT#6,"4 T
EXT: ",A3$(a)
1310 IF RIGHT$(A3$(a),1)=CHR$(32) THEN 132        [2927]
0 ELSE 1430
1320 LINE INPUT#6,"5 TEXT: ",A4$(a)               [2930]
1330 IF RIGHT$(A4$(a),1)=CHR$(32) THEN 134        [3159]
0 ELSE 1430
```

Listing RECHNOMA



Einzelbezug »DATABOX«

☐ Ich mache von Ihrem Angebot Gebrauch und bestelle hiermit die umseitig ausgewählten Produkte.
Bitte liefern Sie die Produkte an die folgende Anschrift:

Name

Straße/Postfach

PLZ/ORT

☐ Den Betrag bezahle ich mittels des beigefügten Verrechnungsschecks.

☐ Ich bitte um Lieferung per Nachnahme zuzüglich der Nachnahmegebühr (nur innerhalb der BRD).

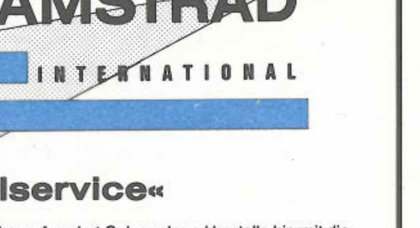
Datum — Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

Bitte ausreichend frankieren

Antwortkarte

DMV-Verlag
PC International
Postfach 250

3440 Eschwege

»Bestellservice«

☐ Ich mache von Ihrem Angebot Gebrauch und bestelle hiermit die umseitig ausgewählten Produkte.
Bitte liefern Sie die Produkte an die folgende Anschrift:

Name

Straße/Postfach

PLZ/ORT

☐ Den Betrag bezahle ich mittels des beigefügten Verrechnungsschecks.

☐ Ich bitte um Lieferung per Nachnahme zuzüglich der Nachnahmegebühr (nur innerhalb der BRD).

Datum — Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

Bitte ausreichend frankieren

Antwortkarte

DMV-Verlag
PC International
Postfach 250

3440 Eschwege



»Kleinanzeigen-Markt«

Absender: *(Bitte genaue Anschrift angeben!)*

Name

Vorname

Firma

Straße/Nr./Postfach

PLZ/Ort

Bitte ausreichend frankieren

Antwortkarte

DMV-Verlag
PC International
Postfach 250

3440 Eschwege



Abo – Order »ZEITSCHRIFT» »DATABOX«

Das kompetente Magazin

Bestellen Sie noch heute Ihr Abonnement mit dieser Postkarte!

Bitte ausreichend frankieren

Antwortkarte

DMV-Verlag
PC International
Postfach 250

3440 Eschwege

Bestellservice

Anz.	Best.-Nr.	Bezeichnung	Preis in DM
		DMV-Software für CPC	
___	202	Copyshop 3"Disk.	49,–
___	103	Startest, Cass.	10,–
___	104	Startest, 3"Disk.	19,–
___	106	Know, 3"Disk.	29,–
___	206	Context CPC, Cass.	49,–
___	207	Context, CPC 3"Disk.	59,–
___	209	FAst BAsic COMpiler	49,–
___	1011	Fantastic Four	29,–
___	213	Software-Experiment, Disk.	39,–
___	210	Fraktal 3D-CPC, Cass.	39,–
___	211	Fraktal 3D-CPC, Disk.	49,–

Anz.	Best.-Nr.	Bezeichnung	Preis in DM
___	1012	Gamebox III	29,–
___	130	Cyrus II-Schach, Cass.	10,–
___	1369	Faszination in 3D, Disk.	39,–
___	205	Special Offer No. 2	59,–
___	107	Special Offer No. 3	39,–
___	1370	CPC-Spielepaket	35,–
___	446	Schneider CPC Assembler-Kurs Buch mit Kassette	39,–
___	447	Buch mit Diskette	49,–
___	481	Schneider CPC-Basic-Trainer 2 Kassetten	29,–
___	214	Turbo-Data-CPC	69,–
___	2902	SuperCalc-2	49,–
___	3511	DMV-Computerwissen, Band 1, Basic-Tools	18,–
___	3512	DMV-Computerwissen DATABOX, 2 Stck. 5 1/4"	35,–
___	3513	DMV-Computerwissen DATABOX, 3 1/2"	35,–

Anz.	Best.-Nr.	Bezeichnung	Preis in DM
		Bücher	
___	401	Prakt. Textverarbeitung mit JOYCE	89,–
___	402	Die BASIC2 TOOLBOX	49,–
___	403	Diskette zum BASIC2-Buch MS-DOS 5 1/4"	29,–
___	404	JOYCE-mehr als ein Textsystem	69,–
___	405	Diskette zum Buch Nr. 404	30,–
___	410	Das große Public-Domain-Buch	34,80
___	412	Das Floppybuch zum CPC	49,–
___	415	Das Maschinensprachebuch zum CPC	39,–
___	417	Das große LOGO-Buch zu CPC und JOYCE	39,–
___	450	Führer zum JOYCE	29,80
___	461	Das BASIC-Buch zum 6128	39,–
___	440	Schneider CPC - Arbeiten mit dBaseII	48,–
___	425	Den JOYCE programmieren	38,–
___	441	Schneider CPC – Arbeiten mit Turbo Pascal	48,–
___	442	Das Schneider CPC Grafikbuch	48,–
___	445	Schneider CPC Erfolg mit Multiplan	48,–

+ Porto/Verpackung
(Inland 4,– DM, Ausland 6,– DM) ______________ DM = Gesamtbetrag ______________ DM

»Einzelheft-/Databox-Bestellung«

Ich bestelle folgende Einzelhefte bzw. Databoxen von »PC Amstrad«:

Einzelheft Ausgabe 6,– DM/Stck	CPC Kassette 14,– DM/Stck.	CPC 3"-Diskette 24,– DM/Stck.	Joyce 3"-Diskette 24,– DM/Stck.	PC 1512 5 1/4"-Diskette 24,– DM/Stck.
☐ 1/88	☐	☐	☐	☐
☐ 2/88	☐	☐	☐	☐
☐ 3/88	☐	☐	☐	☐
☐ 4/88	☐	☐	☐	☐
☐ 5/88	☐	☐	☐	☐
☐ 6/88	☐	☐	☐	☐
☐ 7/88	☐	☐	☐	☐
☐ 8/88	☐	☐	☐	☐
☐ 9/88	☐	☐	☐	☐
☐ 10/88	☐	☐	☐	☐
☐ 11/88	☐	☐	☐	☐
— 12/88	☐	☐	☐	☐
☐ 1/89	☐	☐	☐	☐
☐ 2/89	☐	☐	☐	☐
☐ 3/89	☐	☐	☐	☐
☐ 4/89	☐	☐	☐	☐
☐ 5/89	☐	☐	☐	☐
☐ 6/89	☐	☐	☐	☐
☐ 7/89	☐	☐	☐	☐
☐ 8/89	☐	☐	☐	☐
☐ 9/89	☐	☐	☐	☐
☐ 10/89	☐	☐	☐	☐
☐ 11/89	☐	☐	☐	☐
☐ 12/89	☐	☐	☐	☐
☐ 1/90	☐	☐	☐	☐
— 2/90	☐	☐	☐	☐

CPC-Sonderhefte:

312 ☐ Sonderheft 6/88 .... **14,– DM**
313 ☐ Sonderheft 7/88 .... **14,– DM**
3136 ☐ Sonderheft 8/89 .... **14,– DM**

Sammelordner PC

☐ Best.-Nr. 531 .... 2 Stck. **15,80 DM**

+ Porto/Verpackung
(Inland 4,– DM, Ausland 6,– DM) ______________ DM

Gesamtbetrag: ______________ DM

»Abo-Order Zeitschrift«

Hiermit bestelle ich **»PC International«** für mindestens

☐ 6 Ausgaben ☐ 12 Ausgaben

Preis: **(BRD und West-Berlin)** 12 Ausgaben 66,– DM, 6 Ausgaben 33,– DM.
Auslandspreise: **Europa** 12 Ausgaben 96,– DM, 6 Ausgaben 48,– DM
Außereuropäisches Ausland 12 Ausgaben 120,– DM, 6 Ausgaben 60,– DM
Lieferung erfolgt ab der nächsterreichbaren Ausgabe

Lieferanschrift
☐ Geschäft ☐ Privat

Rechner-Typ
☐ CPC ☐ Joyce ☐ PC

Vorname, Name

Firma (nur wenn Lieferadresse)

Straße, Nr./Postfach

PLZ/Ort *(Bitte genaue Anschrift angeben!)*

Dieses Abonnement verlängert sich automatisch um 6 bzw. 12 Ausgaben, wenn es nicht sechs Wochen vor Ablauf schriftlich gekündigt wird.

Datum Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

Bitte unbedingt zwei Unterschriften leisten!

»Abo-Order DATABOX«

Hiermit bestelle ich die
☐ **CPC Databox** ☐ **Joyce-Databox** ☐ **PC 1512-Databox**
für mindestens ☐ 6 Ausgaben ☐ 12 Ausgaben
als ☐ Cassette ☐ Diskette 3" bzw. 5 1/4".

Bezugspreise Inland (einschl. Porto/Verpackung)
Cass. 6 Ausgaben 90,– DM, 12 Ausgaben 180,– DM
Disk 3" 6 Ausgaben 150,– DM, 12 Ausgaben 300,– DM

Bezugspreise Ausland (eingekl. Werte außereurop.)
Cass. 6 Ausg. 100,– DM (120,–) DM, 12 Ausg. 200,– (240,–) DM
Disk 3" 6 Ausg. 160,– DM (180,–) DM, 12 Ausg. 320,– (360,–) DM

Gewünschte Zahlungsweise:
☐ **Bequem und bargeldlos durch Bankabbuchung**

Bankleitzahl (von Scheck abschreiben)

Konto-Nr./-Inhaber

Geldinstitut

☐ **Gegen Rechnung** — zahlbar innerhalb zwei Wochen nach Erhalt. (Bitte keine Vorauszahlung leisten — Rechnung abwarten.)

Widerrufsrecht:
Ich weiß, daß ich diese Vereinbarung innerhalb einer Woche beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich widerrufen kann, wobei bereits die rechtzeitige Absendung meines Widerrufschreibens zur Fristwahrung ausreicht. Ich bestätige das durch meine zweite Unterschrift.

Datum Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

»Kleinanzeigen-Markt«

Private Anzeigen: Nur DM 5,– je angefangene Zeile.
Geschäftliche Empfehlungen: DM 8,– je angefangene Zeile, zzgl. gesetzlicher Mehrwertsteuer.
Kreuzen Sie bitte an, in welche Rubrik (s. Karte) Ihre Anzeige gehört, schreiben Sie Ihren Text in die Karte (jedes Kästchen = ein Zeichen; Buchstabe, Satzzeichen oder Wortzwischenraum).
Achtung! Der Abdruck erfolgt **nur** gegen Vorkasse (Verrechnungsscheck).
Bitte veröffentlichen Sie meine Anzeige in der nächsterreichbaren **»PC International«** für
☐ private Zwecke ☐ gewerbliche Zwecke (gewerbliche Anzeigen werden mit **G** gekennzeichnet)

Das ist der Text: *(Bitte deutlich in Druckbuchstaben schreiben!)*

☐ Die Anzeige soll als Chiffre-Anzeige erscheinen (nur möglich bei Privat-Anzeige)
Chiffre-Gebühr 10,– DM zzgl. zum Anzeigenpreis

In dieser Rubrik:

☐ **Biete an** ☐ **Suche** ☐ **Tausch** ☐ **Stellenmarkt/freie Mitarbeit**
☐ Hardware ☐ Hardware ☐ **Geschäftsverbindungen**
☐ Software ☐ Software ☐ **Verschiedenes**

Bei Angeboten:
Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze.

Datum Unterschrift (bei Minderjährigen des gesetzl. Vertreters)

```
1340 LINE INPUT#6,"6 TEXT: ",A5$(a)                [3244]
1350 IF RIGHT$(A5$(a),1)=CHR$(32) THEN 136 [2357]
0 ELSE 1430
1360 CLS#6:LOCATE 43,23:PRINT".    n{chste [5536]
 Textzeile  <LEERTASTE>":LINE INPUT#6,"7 T
EXT: ",A6$(a)
1370 IF RIGHT$(A6$(a),1)=CHR$(32) THEN 138 [3591]
0 ELSE 1430
1380 LINE INPUT#6,"8 TEXT: ",A7$(a)                [3107]
1390 IF RIGHT$(A7$(a),1)=CHR$(32) THEN 140 [2530]
0 ELSE 1430
1400 LINE INPUT#6,"9 TEXT: ",A8$(a)                [2659]
1410 IF RIGHT$(A8$(a),1)=CHR$(32) THEN 142 [2704]
0 ELSE 1430
1420 CLS#6:LOCATE 48,23:PRINT CHR$(7);:PRI [6307]
NT "Letzte Textzeile":LINE INPUT#6,"L TEXT
: ",A9$(a)
1430 CLS#6                                         [384]
1440 PRINT#6,"MENGE:"TAB(23)"PREIS:"TAB(48 [2396]
)"PROZENT:"
1450 LOCATE#6,9,1                                  [681]
1460 INPUT#6,;"",M(a)                              [592]
1470 LOCATE#6,31,1                                 [314]
1480 INPUT#6,;"",AP(a):IF AP(a) <> 0 THEN  [2156]
1510
1490 LOCATE#6,58,1                                 [446]
1500 INPUT#6,"",P(a):EP(a)=ROUND(M(a)*AP(a [3687]
),2):GOTO 1730
1510 EP(a)=ROUND(M(a)*AP(a),2)                     [2119]
1520 PRINT#3,A$(a);                                [1063]
1530 IF A1$(a)="" THEN 1710                        [1448]
1540 PRINT#3:PRINT#3,A1$(a);                       [1245]
1550 IF A2$(a)="" THEN 1710                        [1306]
1560 PRINT#3:PRINT#3,A2$(a);                       [1213]
1570 IF A3$(a)="" THEN 1710                        [1047]
1580 PRINT#3:PRINT#3,A3$(a);                       [1297]
1590 IF A4$(a)="" THEN 1710                        [1838]
1600 PRINT#3:PRINT#3,A4$(a);                       [1265]
1610 IF A5$(a)="" THEN 1710                        [987]
1620 PRINT#3:PRINT#3,A5$(a);                       [1120]
1630 IF A6$(a)="" THEN 1710                        [1420]
1640 PRINT#3:PRINT#3,A6$(a);                       [1088]
1650 IF A7$(a)="" THEN 1710                        [988]
1660 PRINT#3:PRINT#3,A7$(a);                       [1440]
1670 IF A8$(a)="" THEN 1710                        [937]
1680 PRINT#3:PRINT#3,A8$(a);                       [1408]
1690 IF A9$(a)="" THEN 1710                        [916]
1700 PRINT#3:PRINT#3,A9$(a);                       [1248]
1710 PRINT#3,TAB(45)USING"#####.##";M(a);: [5124]
PRINT#3,TAB(54)USING"######,.##";AP(a);:PR
INT#3,TAB(69)USING U$;EP(a)
1720 GP=GP+EP(a):GOTO 1960                         [732]
1730 GP=GP+EP(a)                                   [439]
1740 prozent(a)=ROUND(GP/100*p(a),2)               [1281]
1750 GP=GP+prozent(a)                              [892]
1760 PRINT#3,A$(a);                                [1063]
1770 IF A1$(a)="" THEN 1950                        [1305]
1780 PRINT#3:PRINT#3,A1$(a);                       [1245]
1790 IF A2$(a)="" THEN 1950                        [1227]
1800 PRINT#3:PRINT#3,A2$(a);                       [1213]
1810 IF A3$(a)="" THEN 1950                        [968]
1820 PRINT#3:PRINT#3,A3$(a);                       [1297]
1830 IF A4$(a)="" THEN 1950                        [1727]
1840 PRINT#3:PRINT#3,A4$(a);                       [1265]
1850 IF A5$(a)="" THEN 1950                        [1035]
1860 PRINT#3:PRINT#3,A5$(a);                       [1120]
1870 IF A6$(a)="" THEN 1950                        [1404]
1880 PRINT#3:PRINT#3,A6$(a);                       [1088]
1890 IF A7$(a)="" THEN 1950                        [1036]
1900 PRINT#3:PRINT#3,A7$(a)                        [1417]
1910 IF A8$(a)="" THEN 1950                        [1048]
1920 PRINT#3:PRINT#3,A8$(a);                       [1408]
1930 IF A9$(a)="" THEN 1950                        [581]
1940 PRINT#3:PRINT#3,A9$(a);                       [1248]
1950 PRINT#3,TAB(39)USING"###.#";P(a);:PRI [4282]
NT#3,TAB(69)USING U$;prozent(a)
1960 CLS#6:PRINT#6,"<RETURN>          <S> um [1878]
me"
1970 OPT$=CHR$(13)+"s"                             [539]
1980 GOSUB 680                                     [972]
1990 ON K GOTO 2000,2010                           [1679]
2000 NEXT                                          [350]
2010 CLS#6:PRINT#6,"<1> volle Mwst    <2>  [5391]
erm{ssigte Mwst    <3> ohne Mwst"
```

Listing RECHNOMA

```
2020 OPT$="123"                                  [234]
2030 GOSUB 680                                   [972]
2040 Y=K                                         [528]
2050 ON K GOTO 2060,2180,2300                   [1825]
2060 PRINT#3:PRINT#3,STRING$(79,45):PRINT#      [2579]
3
2070 PRINT#3,TAB(49)"NETTO-SUMME      DM"TAB    [2045]
(69)USING u$;GP
2080 PRINT#3                                     [316]
2090 MW=ROUND(GP*STEUER/100,2):BP=GP+MW         [1652]
2100 PRINT#3,TAB(47)"+ ";USING"##.#";STEUE      [5326]
R;:PRINT#3," % MWST      DM"TAB(69)USING U$;
MW
2110 PRINT#3,TAB(64)"---------------"           [1922]
2120 PRINT#3,TAB(49)"T O T A L      DM"TAB      [2799]
(69)USING U$;BP
2130 PRINT#3,TAB(64)"==============="           [1539]
2140 CLS#6:PRINT#6,"<D> rucken      <M> enu     [2375]
e"
2150 OPT$="dm"                                   [441]
2160 GOSUB 680                                   [972]
2170 ON K GOTO 2370,3660                        [1638]
2180 PRINT#3:PRINT#3,STRING$(79,45):PRINT#      [2579]
3
2190 PRINT#3,TAB(49)"NETTO-SUMME      DM"TAB    [2045]
(69)USING u$;GP
2200 PRINT#3                                     [316]
2210 MW=ROUND(GP*STEUER1/100,2):BP=GP+MW        [1250]
2220 PRINT#3,TAB(47)"+ ";USING"##.#";STEUE      [5240]
R1;:PRINT#3," % MWST      DM"TAB(69)USING U$
;MW
2230 PRINT#3,TAB(64)"---------------"           [1922]
2240 PRINT#3,TAB(49)"T O T A L      DM"TAB      [2799]
(69)USING U$;BP
2250 PRINT#3,TAB(64)"==============="           [1539]
2260 CLS#6:PRINT#6,"<D> rucken      <M> enu     [2375]
e"
2270 OPT$="dm"                                   [441]
2280 GOSUB 680                                   [972]
2290 ON K GOTO 2370,3660                        [1638]
2300 PRINT#3:PRINT#3,STRING$(79,45):PRINT#      [2579]
3
2310 PRINT#3,TAB(51)"T O T A L      DM"TAB(    [2221]
69)USING U$;GP
2320 PRINT#3,TAB(64)"==============="           [1539]
2330 CLS#6:PRINT#6,"<D> rucken      <M> enu     [2375]
e"
2340 OPT$="dm"                                   [441]
2350 GOSUB 680                                   [972]
2360 ON K GOTO 2590,3660                        [1455]
2370 PRINT#8,CHR$(27);CHR$(64);:CLS#6           [2438]
2380 PRINT#8,CHR$(27);CHR$(108);CHR$(6);        [2282]
2390 PRINT#6,"Anzahl Ausdrucke  <4>"            [2085]
2400 LOCATE#6,20,1:INPUT#6,"",Anzahl            [1782]
2410 IF Anzahl <> 0 THEN 2430                   [1021]
2420 Anzahl=4                                    [188]
2430 FOR Ausdruck=1 TO Anzahl                   [1226]
2440 GOSUB 2760                                  [899]
2450 PRINT#8,STRING$(70,45)                     [1671]
2460 PRINT#8:PRINT#8,TAB(42)"NETTO-SUMME        [3886]
 DM"TAB(60)USING U$;GP
2470 PRINT#8                                     [343]
2480 IF Y=2 THEN 2500                            [865]
2490 PRINT#8,TAB(40);"+ ";USING"##.#";STEU      [5656]
ER;:PRINT#8," % MWST     DM"TAB(60)USING U$;
MW:GOTO 2510
2500 PRINT#8,"TAB(40);"+ ";USING"##.#";STEU     [5356]
ER1;:PRINT#8," % MWST     DM"TAB(60)USING U$
;MW
2510 PRINT#8,TAB(56)"---------------"           [1606]
2520 PRINT#8,CHR$(27);CHR$(71);:PRINT#8,TA      [4731]
B(43)"T O T A L      DM"TAB(61)USING U$;BP
2530 PRINT#8,CHR$(27);CHR$(72);                 [1938]
2540 PRINT#8,TAB(57)"==============="           [1895]
2550 PRINT#8,CHR$(12);                          [1101]
2560 GP1=0                                       [363]
2570 NEXT                                        [350]
2580 GOTO 2140                                   [351]
2590 PRINT#8,CHR$(27);CHR$(64);:CLS#6           [2438]
2600 PRINT#8,CHR$(27);CHR$(108);CHR$(6);        [2282]
2610 PRINT#6,"Anzahl Ausdrucke  <4>"            [2085]
2620 LOCATE#6,20,1:INPUT#6,"",Anzahl            [1782]
2630 IF Anzahl <> 0 THEN 2650                    [977]
2640 Anzahl=4                                    [188]
2650 FOR Ausdruck=1 TO Anzahl                   [1226]
```

Listing RECHNOMA

```
2660 GOSUB 2760                                  [899]
2670 PRINT#8,STRING$(70,45)                     [1671]
2680 PRINT#8,CHR$(27);CHR$(71);                 [1942]
2690 PRINT#8:PRINT#8,TAB(42)"T O T A L          [3154]
 DM"TAB(60)USING U$;GP
2700 PRINT#8,CHR$(27);CHR$(72);                 [1938]
2710 PRINT#8,TAB(57)"==============="           [1895]
2720 PRINT#8,CHR$(12);                          [1101]
2730 GP1=0                                       [363]
2740 NEXT                                        [350]
2750 GOTO 2330                                   [345]
2760 IF druck=1 THEN GOSUB 5550:GOTO 2850       [1704]
2770 PRINT#8:PRINT#8:PRINT#8:PRINT#8:PRINT      [2336]
#8
2780 PRINT#8,CHR$(27)+CHR$(69);TAB(48)"Dat      [3799]
um: ";CHR$(27)+CHR$(70);D$
2790 PRINT#8:PRINT#8:PRINT#8                    [1827]
2800 PRINT#8,ANR$                                [843]
2810 PRINT#8,NAM$                               [1295]
2820 PRINT#8,STRASS$                            [1532]
2830 PRINT#8                                     [343]
2840 PRINT#8,ORT$                               [1141]
2850 PRINT#8,CHR$(27);CHR$(69);                 [1869]
2860 PRINT#8:PRINT#8:PRINT#8:PRINT#8:PRINT      [4089]
#8:PRINT#8:PRINT#8:PRINT#8,R$
2870 PRINT#8,CHR$(27);CHR$(70);                 [1946]
2880 PRINT#8:PRINT#8,STRING$(70,45)             [2291]
2890 PRINT#8,b$                                  [880]
2900 IF b1$="" THEN 2920                        [1260]
2910 PRINT#8,b1$                                 [907]
2920 IF b2$="" THEN 2940                        [1064]
2930 PRINT#8,b2$                                 [913]
2940 IF b3$="" THEN 2960                         [775]
2950 PRINT#8,b3$                                 [903]
2960 PRINT#8,STRING$(70,45)                     [1671]
2970 Seite=1                                     [882]
2980 PRINT#8:PRINT#8,TAB(36)"%"TAB(42)"MEN      [4751]
GE"TAB(53)"PREIS"TAB(62)"BETRAG DM":PRINT#
8
2990 Zeile=31                                    [679]
3000 FOR a=1 TO 100                              [909]
3010 IF Zeile>55 THEN GOSUB 5800                [1451]
3020 IF M(a)=0 AND EP(a)=0 AND PROZENT(a)=      [2829]
0 THEN RETURN
3030 IF AP(a)=0 THEN 3340                       [1032]
3040 PRINT#8,A$(a);                             [1164]
3050 Zeile=Zeile+1                               [564]
3060 IF A1$(a)="" THEN 3330                     [1428]
3070 Zeile=Zeile+1                               [564]
3080 PRINT#8:PRINT#8,A1$(a);                    [1742]
3090 IF A2$(a)="" THEN 3330                     [1078]
3100 Zeile=Zeile+1                               [564]
3110 PRINT#8:PRINT#8,A2$(a);                    [1965]
3120 IF A3$(a)="" THEN 3330                      [901]
3130 Zeile=Zeile+1                               [564]
3140 PRINT#8:PRINT#8,A3$(a);                    [1806]
3150 IF A4$(a)="" THEN 3330                     [1930]
3160 Zeile=Zeile+1                               [564]
3170 PRINT#8:PRINT#8,A4$(a);                    [1772]
3180 IF A5$(a)="" THEN 3330                     [1015]
3190 Zeile=Zeile+1                               [564]
3200 PRINT#8:PRINT#8,A5$(a);                    [1867]
3210 IF A6$(a)="" THEN 3330                     [1384]
3220 Zeile=Zeile+1                               [564]
3230 PRINT#8:PRINT#8,A6$(a);                    [2091]
3240 IF A7$(a)="" THEN 3330                     [1016]
3250 Zeile=Zeile+1                               [564]
3260 PRINT#8:PRINT#8,A7$(a);                    [1933]
3270 IF A8$(a)="" THEN 3330                      [917]
3280 Zeile=Zeile+1                               [564]
3290 PRINT#8:PRINT#8,A8$(a);                    [1899]
3300 IF A9$(a)="" THEN 3330                      [769]
3310 Zeile=Zeile+1                               [564]
3320 PRINT#8:PRINT#8,A9$(a);                    [1753]
3330 PRINT#8,TAB(39)USING"#####.##";M(a);:      [7006]
PRINT#8,TAB(48)USING"######,.##";AP(a);:PR
INT#8,TAB(60)USING U$;EP(a):GOTO 3640
3340 PRINT#8,A$(a);                             [1164]
3350 Zeile=Zeile+1                               [564]
3360 IF A1$(a)="" THEN 3630                     [1281]
3370 Zeile=Zeile+1                               [564]
3380 PRINT#8:PRINT#8,A1$(a);                    [1742]
3390 IF A2$(a)="" THEN 3630                     [1410]
3400 Zeile=Zeile+1                               [564]
3410 PRINT#8:PRINT#8,A2$(a);                    [1965]
3420 IF A3$(a)="" THEN 3630                     [1071]
```

Listing RECHNOMA

```
3430 Zeile=Zeile+1                              [564]
3440 PRINT#8:PRINT#8,A3$(a);                    [1806]
3450 IF A4$(a)="" THEN 3630                     [1894]
3460 Zeile=Zeile+1                              [564]
3470 PRINT#8:PRINT#8,A4$(a);                    [1772]
3480 IF A5$(a)="" THEN 3630                     [835]
3490 Zeile=Zeile+1                              [564]
3500 PRINT#8:PRINT#8,A5$(a);                    [1867]
3510 IF A6$(a)="" THEN 3630                     [1332]
3520 Zeile=Zeile+1                              [564]
3530 PRINT#8:PRINT#8,A6$(a);                    [2091]
3540 IF A7$(a)="" THEN 3630                     [836]
3550 Zeile=Zeile+1                              [564]
3560 PRINT#8:PRINT#8,A7$(a);                    [1933]
3570 IF A8$(a)="" THEN 3630                     [1024]
3580 Zeile=Zeile+1                              [564]
3590 PRINT#8:PRINT#8,A8$(a);                    [1899]
3600 IF A9$(a)="" THEN 3630                     [924]
3610 Zeile=Zeile+1                              [564]
3620 PRINT#8:PRINT#8,A9$(a);                    [1753]
3630 PRINT#8,TAB(32)USING"###.#";P(a);:PRI      [4473]
NT#8,TAB(60)USING U$;PROZENT(a):Zeile=Zeil
e+1
3640 GP1=GP1+EP(a)+PROZENT(a):NEXT              [2385]
3650 RETURN                                     [555]
3660 ERASE A$,A1$,A2$,A3$,A4$,A5$,A6$,A7$,      [2196]
A8$,A9$,M,AP,EP,P,prozent
3670 DIM A$(100),A1$(100),A2$(100),A3$(100      [6615]
),A4$(100),A5$(100),A6$(100),A7$(100),A8$(
100),A9$(100),M(100),AP(100),EP(100),P(100
),prozent(100)
3680 GOTO 490                                   [371]
3690 CLS#4                                      [386]
3700 MOVE 155,140:DRAWR 0,119:DRAWR 330,0:      [3191]
DRAWR 0,-119:DRAWR -330,0
3710 LOCATE#1,1,1:PRINT#1,"1.Stammnummer"       [2760]

3720 PRINT#1                                    [318]
3730 PRINT#1,"2.Anrede"                         [1034]
3740 PRINT#1,"3.Vorname, Name"                  [1133]

3750 PRINT#1,"4.Strasse":PRINT#1                [1667]
3760 PRINT#1,"5.PLZ / Wohnort"                  [2050]

3770 CLS#2                                      [372]
3780 INPUT#2," ",such$                          [1514]
3790 FOR x=1 TO HH                              [829]
3800 IF x=HH AND INSTR(stamm$(x),such$)=0       [2888]
THEN 3890
3810 IF INSTR(stamm$(x),such$)=0 THEN 3880      [1762]
3820 PRINT#2                                    [317]
3830 PRINT#2," ";ANR$(x)                        [758]
3840 PRINT#2," ";Nam$(x)                        [578]
3850 PRINT#2," ";strass$(x)                     [1188]
3860 PRINT#2:PRINT#2," ";Ort$(x)                [1644]
3870 GOTO 3900                                  [493]
3880 NEXT                                       [350]
3890 LOCATE 31,24:PRINT CHR$(7)" Satz nich      [7524]
t in Datei ":FOR i=1 TO 1000:NEXT:LOCATE 3
1,24:PRINT CHR$(24)"   Stammdaten {ndern  "
CHR$(24):GOTO 3770
3900 stamm$=stamm$(x):anr$=anr$(x):nam$=na      [7414]
m$(x):strass$=strass$(x):ort$=ort$(x)
3910 RETURN                                     [555]
3920 LOCATE 30,5:PRINT CHR$(24)"  S t a m       [4076]
m d a t e n  "CHR$(24)
3930 LOCATE 35,11:PRINT"1 - Anlegen     "       [2061]
3940 LOCATE 35,12:PRINT"2 - Anzeige     "       [2329]
3950 LOCATE 35,13:PRINT"3 - Drucken     "       [1778]
3960 LOCATE 35,14:PRINT"4 - [ndern      "       [2252]
3970 LOCATE 35,16:PRINT"5 - Hauptmenue":LO      [6274]
CATE 31,24:PRINT CHR$(24)"   Bitte w{hlen S
ie:  "CHR$(24)
3980 opt$="12345"                               [381]
3990 GOSUB 680                                  [972]
4000 ON K GOSUB 4020,4340,4570,4720,580         [2076]
4010 CLS#4:GOTO 3920                            [1476]
4020 LOCATE 31,24:PRINT CHR$(24)"   Stammda     [2919]
ten anlegen  "CHR$(24)
4030 CLS#4:CLS#2                                [794]
4040 MOVE 155,140:DRAWR 0,119:DRAWR 330,0:      [3191]
DRAWR 0,-119:DRAWR -330,0
4050 LOCATE 17,8:PRINT"Letzter Datensatz:       [6059]
      Freie Datens{tze:"
4060 LOCATE#1,1,1                               [571]
4070 PRINT#1,"1.Stammnummer"                    [1844]
4080 PRINT#1                                    [318]
4090 PRINT#1,"2.Anrede"                         [1034]
4100 PRINT#1,"3.Vorname, Name"                  [1133]
```

Listing RECHNOMA

```
4110 PRINT#1,"4.Strasse"                        [1377]
4120 PRINT#1,"5.PLZ / Wohnort"                  [2050]
4130 LOCATE 21,18:PRINT"Abbruch durch Leer      [3142]
eingabe Zeile 2 und 3"
4140 FOR x=1 TO HH                              [829]
4150 Satz=Satz+1                                [1147]
4160 frei=HH-Satz                               [208]
4170 IF LEN(stamm$(x))>0 THEN 4270              [2054]
4180 CLS#2:Stamm$(x)=STR$(Satz)                 [2241]
4190 PRINT#2," ";Stamm$(x)                      [1341]
4200 PRINT#2                                    [317]
4210 LINE INPUT#2," ",Anr$(x)                   [1605]
4220 LINE INPUT#2," ",Nam$(x)                   [1955]
4230 IF Anr$(x)="" AND Nam$(x)="" THEN 431      [2122]
0
4240 LINE INPUT#2," ",strass$(x)                [2315]
4250 LINE INPUT#2," ",ort$(x)                   [1461]
4260 CLS#2                                      [372]
4270 Satz=VAL(Stamm$(x))                        [1269]
4280 Satz(x)=VAL(stamm$(x))                     [2048]
4290 LOCATE 35,8:PRINT USING"####";Satz(x)      [4834]
:LOCATE 62,8:PRINT USING"####";frei
4300 NEXT                                       [350]
4310 GOSUB 4480                                 [1017]
4320 CLS#4                                      [386]
4330 RETURN                                     [555]
4340 CLS#4:PRINT#4                              [968]
4350 FOR x=1 TO HH                              [829]
4360 IF stamm$(x)="" THEN 4450                  [1069]
4370 IF x=6 OR x=11 OR x=16 OR x=21 OR x=2      [8334]
6 OR x=31 OR x=36 OR x=41 OR x=46 OR x=51
OR x=56 OR x=61 OR x=66 OR x=71 OR x=76 OR
 x=81 OR x=86 OR x=91 OR x=96 THEN GOSUB 4
940
4380 PRINT#4,stamm$(x);                         [1283]
4390 PRINT#4,TAB(5);anr$(x);                    [1552]
4400 IF anr$(x)="" THEN PRINT#4,TAB(5)nam$      [5971]
(x) ELSE PRINT#4,TAB(41)nam$(x);
4410 PRINT#4,TAB(5)strass$(x);                  [1111]
4420 PRINT#4,TAB(41)ort$(x)                     [1281]
4430 PRINT#4                                    [315]
4440 NEXT                                       [350]
4450 LOCATE 31,24:PRINT CHR$(24)" D a t e       [2133]
   E n d e "CHR$(24)
4460 CALL &BB18                                 [389]
4470 RETURN                                     [555]
4480 stamm$(x)=stamm$:anr$(x)=anr$:nam$(x)      [5385]
=nam$:strass$(x)=strass$:ort$(x)=ort$
4490 stamm$="":anr$="":nam$="":strass$="":      [3582]
ort$=""
4500 CLS#4                                      [386]
4510 OPENOUT"ADRESSEN.DAT"                      [1617]
4520 FOR x=1 TO HH                              [829]
4530 WRITE#9,stamm$(x),Anr$(x),nam$(x),str      [3602]
ass$(x),ort$(x)
4540 NEXT                                       [350]
4550 CLOSEOUT                                   [902]
4560 RETURN                                     [555]
4570 Seite=1                                    [882]
4580 PRINT#8,TAB(9)STRING$(70,"*")              [1866]
4590 PRINT#8,TAB(9)"* Adressenliste             [5390]
          - Seite";Seite"         "
D$" *"
4600 PRINT#8,TAB(9)STRING$(70,"*"):PRINT#8      [2874]
4610 FOR x=1 TO HH                              [829]
4620 IF stamm$(x)="" THEN 4700                  [1051]
4630 IF x=21 OR x=41 OR x=61 OR x=81 THEN       [2513]
GOSUB 5980
4640 PRINT#8,TAB(4)stamm$(x);                   [1270]
4650 PRINT#8,TAB(9)anr$(x);                     [1697]
4660 IF anr$(x)="" THEN PRINT#8,TAB(9)nam$      [5530]
(x) ELSE PRINT#8,TAB(45)nam$(x);
4670 PRINT#8,TAB(9)strass$(x);                  [1889]
4680 PRINT#8,TAB(45)ort$(x)                     [1667]
4690 PRINT#8,TAB(9)STRING$(70,"-")              [2189]
4700 NEXT                                       [350]
4710 RETURN                                     [555]
4720 LOCATE 31,24:PRINT CHR$(24)"   Stammda     [3667]
ten {ndern  "CHR$(24)
4730 GOSUB 3690:GOTO 4750                       [1827]
4740 GOSUB 4830                                 [956]
4750 LOCATE 30,19:PRINT STRING$(41,32)          [1868]
4760 LOCATE 4,19:INPUT"Welche Zeile ? (6 =      [4707]
 Ende) ",zeile
4770 IF zeile<2 OR zeile>6 THEN PRINT CHR$      [6221]
(7):LOCATE 30,19:PRINT CHR$(24)" unzul{ssi
ge Eingabe ! "CHR$(24):FOR a=1 TO 1000:NEX
T:GOTO 4750
4780 ON zeile GOTO 4770,4790,4800,4810,482      [2773]
0,4920
```

Listing RECHNOMA

```
4790 LOCATE 30,19:PRINT CHR$(24)" 2. Anred [4727]
e:"CHR$(24)" ";:LINE INPUT"",Anr$:GOTO 47
40
4800 LOCATE 30,19:PRINT CHR$(24)" 3. Vorna [5182]
me, Name:"CHR$(24)" ";:LINE INPUT"", Nam$
: GOTO 4740
4810 LOCATE 30,19:PRINT CHR$(24)" 4. Stras [4130]
se:"CHR$(24)" ";:LINE INPUT"",strass$: GO
TO 4740
4820 LOCATE 30,19:PRINT CHR$(24)" 5. PLZ / [5692]
 Wohnort:"CHR$(24)" ";:LINE INPUT"",ort$:
GOTO 4740
4830 CLS#2                                 [372]
4840 PRINT#2," ";stamm$                    [1250]
4850 PRINT#2                               [317]
4860 PRINT#2," ";Anr$                      [757]
4870 PRINT#2," ";nam$                      [579]
4880 PRINT#2," ";strass$                   [1063]
4890 PRINT#2                               [317]
4900 PRINT#2," ";ort$                      [800]
4910 RETURN                                [555]
4920 GOSUB 4480                            [1017]
4930 CLS#4:GOTO 3920                       [1476]
4940 LOCATE 31,24:PRINT CHR$(24)" <RETURN> [4001]
 <"CHR$(240)"> Ende "CHR$(24)
4950 a$=INKEY$:IF a$="" THEN 4950          [1538]
4960 IF a$=CHR$(240) THEN CLS#4:GOTO 3920  [1000]
4970 IF a$=CHR$(13) THEN 4980 ELSE 4950    [601]
4980 CLS#4:PRINT#4                         [968]
4990 RETURN                                [555]
5000 LOCATE 30,5:PRINT CHR$(24)"   S t e u [4221]
e r s a t z "CHR$(24)
5010 LOCATE 29,11:PRINT STRING$(20,32)     [1851]
5020 LOCATE 29,12:PRINT STRING$(20,32)     [1272]
5030 LOCATE 29,13:PRINT"Voller Steuersatz: [3995]
   %"
5040 LOCATE 29,14:PRINT"Erm{ss.Steuersatz: [3325]
   %"
5050 LOCATE 29,15:PRINT STRING$(20,32)     [1527]
5060 LOCATE 29,16:PRINT STRING$(20,32):LOC [4558]
ATE 31,24:PRINT CHR$(24)" Bitte w{hlen Si
e: "CHR$(24)
5070 LOCATE 31,24:PRINT CHR$(24)" Steuers  [3863]
atz {ndern "CHR$(24)
5080 LOCATE 48,13:INPUT"",STEUER           [1479]
5090 LOCATE 49,14:INPUT"",STEUER1          [2179]
5100 OPENOUT"MWST.DAT"                     [915]
5110 WRITE#9,STEUER,STEUER1                [2979]
5120 CLOSEOUT                              [902]
5130 GOTO 580                              [448]
5140 CLS:PRINT CHR$(7):LOCATE 20,10:PRINT" [6527]
 Programm muss erst installiert werden! "
5150 LOCATE 21,12:PRINT" Programm "CHR$(24 [5359]
)" INSTALL.BAS "CHR$(24)" wird geladen!"
5160 FOR i=1 TO 1000:NEXT                  [741]
5170 RUN "INSTALL.BAS"                     [1794]
5180 REM                                   [272]
5190 LOCATE 30,5:PRINT CHR$(24)"   K o p f [2437]
 z e i l e "CHR$(24)
5200 CLS#4                                 [386]
5210 PRINT CHR$(7):LOCATE#4,14,8:PRINT#4," [6213]
Vorsicht, bestehende Datei wird }berschrie
ben!!"
5220 LOCATE#4,23,10:PRINT#4,"<M>enue    <  [4530]
RETURN> = weiter"
5230 opt$="m"+CHR$(13)                     [584]
5240 GOSUB 680                             [972]
5250 ON K GOTO 5260,5270                   [1932]
5260 CLS#4:GOTO 580                        [1430]
5270 CLS#4                                 [386]
5280 LOCATE#4,1,5                          [360]
5290 PRINT#4,TAB(10);"Firma 1. Zeile   "CHR [2756]
$(246)TAB(52)CHR$(247)
5300 PRINT#4,TAB(10);"        2. Zeile   "CHR [2780]
$(246)TAB(52)CHR$(247)
5310 PRINT#4,TAB(10);"Stra~e             "CHR [2295]
$(246)TAB(52)CHR$(247)
5320 PRINT#4,TAB(10);"PLZ, Ort           "CHR [4008]
$(246)TAB(52)CHR$(247)
5330 PRINT#4,TAB(10);"Telefon            "CHR [3262]
$(246)TAB(52)CHR$(247)
5340 PRINT#4,TAB(10);"Bankverbindung     "CHR [3696]
$(246)TAB(52)CHR$(247)
5350 PRINT#4,TAB(10);"Konto-Nr.          "CHR [2997]
$(246)TAB(52)CHR$(247)
5360 PRINT#4,TAB(10);"Bankleitzahl       "CHR [2688]
$(246)TAB(52)CHR$(247)
5370 LOCATE 31,24:PRINT CHR$(24)" Kopfzei  [4111]
le anlegen  "CHR$(24)
```

Listing RECHNOMA

```
5380 LOCATE 30,11:LINE INPUT"",Vorname$    [1933]
5390 LOCATE 30,12:LINE INPUT"",Name$       [1349]
5400 LOCATE 30,13:LINE INPUT"",Strasse$    [1484]
5410 LOCATE 30,14:LINE INPUT"",Plz$        [1265]
5420 LOCATE 30,15:LINE INPUT"",Telefon$    [1326]
5430 LOCATE 30,16:LINE INPUT"",Bank$       [2206]
5440 LOCATE 30,17:LINE INPUT"",Kto$        [1684]
5450 LOCATE 30,18:LINE INPUT"",Blz$        [1692]
5460 CLS#4                                 [386]
5470 OPENOUT"KOPF.DAT"                     [1149]
5480 WRITE#9,Vorname$,Name$,Strasse$,Plz$, [2780]
Telefon$,Bank$,Kto$,Blz$
5490 CLOSEOUT                              [902]
5500 GOTO 580                              [448]
5510 OPENIN"KOPF.DAT"                      [1610]
5520 INPUT#9,Vorname$,Name$,Strasse$,Plz$, [3466]
Telefon$,Bank$,Kto$,Blz$
5530 CLOSEIN                               [752]
5540 GOTO 490                              [371]
5550 PRINT#8,CHR$(27)+CHR$(69)             [1218]
5560 laenge=LEN(Vorname$)*2                [989]
5570 abstand=(48-laenge)-1                 [1849]
5580 PRINT#8,CHR$(27)+CHR$(87)+CHR$(49)Vor [6776]
name$+CHR$(27)+CHR$(87)+CHR$(48);SPC(absta
nd)Plz$
5590 laenge1=LEN(Name$)                    [1531]
5600 mitte=laenge-laenge1                  [1056]
5610 abstand1=INT(mitte/2)                 [1412]
5620 PRINT#8,TAB(abstand1)Name$;TAB(48)str [2541]
asse$
5630 IF Telefon$="" THEN GOTO 5650         [2018]
5640 PRINT#8,TAB(48)"Telefon: ";Telefon$   [2269]
5650 IF Bank$="" THEN PRINT#8:GOTO 5700    [1782]
5660 PRINT#8:PRINT#8,TAB(48)"Bankverbindun [4126]
g:"
5670 PRINT#8,TAB(48)Bank$                  [1719]
5680 PRINT#8,TAB(48)"Konto:  "Kto$         [1613]
5690 PRINT#8,TAB(48)"BLZ      "Blz$:GOTO 571 [2051]
0
5700 PRINT#8,TAB(48)"Datum:  "CHR$(27)+CHR$ [5690]
(70)D$:PRINT#8:PRINT#8:PRINT#8:GOTO 5720
5710 PRINT#8,CHR$(27)+CHR$(70);            [1273]
5720 IF Telefon$="" THEN PRINT#8           [1783]
5730 PRINT#8,ANR$                          [843]
5740 PRINT#8,NAM$;:IF Bank$<>"" THEN PRINT [5939]
#8,TAB(48)CHR$(27)+CHR$(69)"Datum: "CHR$(2
7)+CHR$(70)D$
5750 IF Bank$="" THEN PRINT#8              [985]
5760 PRINT#8,Strass$                       [1532]
5770 PRINT#8                               [343]
5780 PRINT#8,ORT$                          [1141]
5790 RETURN                                [555]
5800 PRINT#8,STRING$(70,45)                [1671]
5810 PRINT#8,TAB(50)"]bertrag:"TAB(60);USI [4129]
NG u$;GP1
5820 PRINT#8,CHR$(12)                      [1074]
5830 Seite=Seite+1                         [932]
5840 PRINT#8,CHR$(27)+CHR$(69);            [1326]
5850 IF druck=2 THEN GOSUB 5950:GOTO 5880  [1857]
5860 PRINT#8,CHR$(27)+CHR$(87)+CHR$(49)Vor [8946]
name$+CHR$(27)+CHR$(87)+CHR$(48);SPC(absta
nd)"Datum: ";CHR$(27)+CHR$(70);D$
5870 PRINT#8,CHR$(27)+CHR$(69)TAB(abstand1 [3690]
)Name$;CHR$(27)+CHR$(70);TAB(57)"- Seite "
Seite;"-"
5880 PRINT#8:PRINT#8:PRINT#8:PRINT#8       [1873]
5890 PRINT#8,TAB(36)"%"TAB(42)"MENGE"TAB(5 [3513]
3)"PREIS"TAB(62)"BETRAG DM"
5900 PRINT#8,STRING$(70,45)                [1671]
5910 PRINT#8,TAB(50)"]bertrag:"TAB(60);USI [4129]
NG u$;GP1
5920 PRINT#8                               [343]
5930 Zeile=9                               [508]
5940 RETURN                                [555]
5950 PRINT#8,TAB(49)"Datum: ";CHR$(27)+CHR [2534]
$(70);D$
5960 PRINT#8,TAB(55)"- Seite "Seite;"-"    [2623]
5970 RETURN                                [555]
5980 PRINT#8,CHR$(12)                      [1074]
5990 Seite=Seite+1                         [932]
6000 PRINT#8,TAB(9)STRING$(70,"*")         [1866]
6010 PRINT#8,TAB(9)"* Adressenliste        [5390]
                     - Seite";Seite"-   "
D$" *"
6020 PRINT#8,TAB(9)STRING$(70,"*")         [1866]
6030 PRINT#8                               [343]
6040 RETURN                                [555]
```

Listing RECHNOMA

Sly Sprite

Strategie für Zwei

Die schönsten Spiele sind die, bei denen man sich mit einem menschlichen Gegner auseinandersetzen kann. Da ist viel Raum für Ärger und Aufregung vorhanden. Man kann sich nach allen Regeln der Kunst "anmaulen" und ärgern. Nach dem Spiel ist dann die Welt wieder in Ordnung, und man fühlt sich gleich viel entspannter.

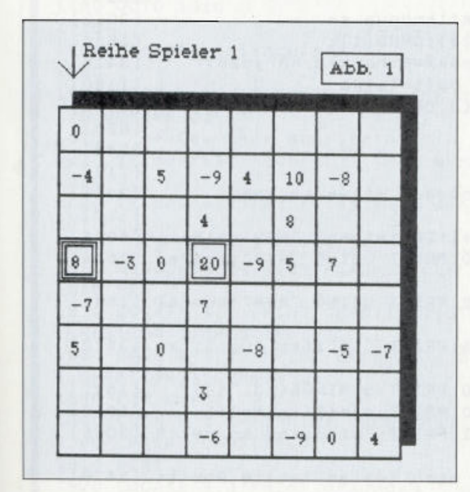


Einfaches Spielprinzip, hohe taktische Anforderungen: So präsentiert sich Sly Sprite

Bei Sly Sprite handelt es sich um eine Mischung aus Glücks- und Strategiespiel. Nach Eingabe der Namen (maximal 13 Buchstaben lang) wird das Spielfeld nebst Informations-Window aufgebaut.

Das Spielfeld ist quadratisch und besteht aus 64 Feldern. Der Computer füllt diese Felder mit Zahlen, die Werte zwischen 20 und −15 annehmen können. Im Informations-Window sind die Punktezahlen der Gegner und der Name des Spielers, der an der Reihe ist, einzusehen.

Die beiden Spieler können per Spacetaste oder Joystickbewegung einen Cursor über das Spielfeld bewegen. Spieler 1 kann den Cursor nur senkrecht bewegen, Spieler 2 nur waagerecht. Der Spieler, der an der Reihe ist, bewegt nun den Cursor über die aktuelle Reihe, bis er an einer Zahl angelangt ist, die ihm paßt. Drückt er nun die Enter-Taste, so wird das Feld, auf dem er den Cursor plaziert hat, gelöscht, und der Wert der dort stehenden Zahl wird seinem Punktekonto gutgeschrieben oder abgezogen.



Taktische Zahlenjagd

Jeder Spieler ist natürlich bemüht, die höchsten Werte "abzusahnen", doch fängt genau hier das taktische Moment des Spiels an. Beispiel:

Spieler 1 (senkrechte Reihen) ist am Zug. Beschließt er, die 8 zu nehmen (Abbildung 1), so hat er zwar den höchsten Wert seiner Reihe bekommen, ermöglicht aber seinem Gegner (waagerechte Reihen), die 20 abzuräumen. Würde er statt dessen die um 3 Punkte geringere 5 (Abbildung 2) nehmen, so müßte sich sein Kontrahent eine negative Zahl gutschreiben.

Das hier in aller Einfachheit dargestellte Spielverhalten kann natürlich viel weiter vertieft werden, so daß Spielzüge – fast wie beim Schach – auf einige Züge im voraus geplant werden müssen.

So wird Sly Sprite zu einer echten Herausforderung für die kleinen grauen Zellen.

(Michael Lona/jf)

Reihe Spieler 1 — Abb. 2

0							
-4		5	-9	4	10	-8	
			4		8		
8	-3	0	20	-9	5	7	
-7			7				
5		0		-8		-5	-7
			3				
			-6		-9	0	4

BASIC-Programme gesucht!

Für unsere ständige PC-Rubrik suchen wir BASIC2-Programme sowie Tips & Tricks.
Alles, was Sie tun müssen, ist, Ihr selbstgeschriebenes Programm mit einer Bedienungsanleitung als Textdatei auf Diskette zu speichern und uns diese zuzusenden.
Als Lohn für Ihre Mühe winkt bei Veröffentlichung ein interessantes Honorar.
Übrigens liegen die besten Programme meist in den Schubladen (wo sie absolut nichts zu suchen haben) und werden aus fehlender Überzeugung nicht eingesandt.
Da wir grundsätzlich jedes Programm ausführlich begutachten, könnte Ihre Einsendung, versehen mit unseren Verbesserungsvorschlägen, vielleicht der Hit des nächsten Monats werden.
Also, auf bald....

Einsendungen bitte an den
DMV-Verlag · PC-Redaktion
Postfach 250 · 3440 Eschwege

für 464-664-6128

```
10 ' ********************************  [1391]
20 ' **      S L Y  S P R I T E    **  [593]
30 ' **                            **  [104]
40 ' **   FUER  CPC 464/664/6128   **  [1106]
50 ' ********************************  [1391]
60 ' **                            **  [104]
70 ' ** Autor: Michael Lona        **  [1169]
80 ' ** Enstehungszeit: Maerz 1989 **  [1571]
90 ' **                            **  [104]
95 ' ********************************  [1391]
100 '                                  [117]
110 ' *** Vorbereitungen ***           [1007]
120 '                                  [117]
130 SYMBOL 250,66,162,34,34,34,34,34,29  [2242]
140 SYMBOL 251,60,98,162,36,56,36,34,97  [2310]
150 SYMBOL 252,97,162,34,36,56,40,36,195 [2125]
160 SYMBOL 253,97,178,50,42,42,38,166,66 [2651]
170 SYMBOL 254,60,98,161,33,33,97,162,92 [1861]
180 SYMBOL 255,28,34,64,64,48,64,66,60   [2605]
190 urkunde$=CHR$(250)+CHR$(251)+CHR$(252) [4764]
+CHR$(250)+CHR$(253)+CHR$(254)+CHR$(255)
200 DIM feld(8,8),score(2),name$(2)     [2845]
210 ENT 1,9,-10,5                       [538]
220 CALL &BC00:PEN 1:PAPER 0:MODE 1     [926]
230 LOCATE 3,12:PRINT "Benoetigen Sie eine [3624]
 Anleitung (J/N)?"
240 e$=UPPER$(INKEY$):IF e$="" THEN 240 [2211]
250 IF e$="J" THEN GOTO 2160            [873]
260 IF e$="N" THEN 270 ELSE 240         [890]
270 MODE 2                              [513]
280 LOCATE 1,1:INPUT "Name des ersten Spie [4918]
lers (senkrecht) : ",name$(1)
290 IF LEN(name$(1))<1 OR LEN(name$(1))>13 [2330]
 THEN 280
300 LOCATE 1,3:INPUT "Name des zweiten Spi [4748]
elers (waagrecht): ",name$(2)
310 IF LEN(name$(2))<1 OR LEN(name$(2))>13 [2996]
 THEN 300
320 '                                   [117]
330 ' *** Spielfeldaufbau ***           [1412]
340 '                                   [117]
350 CLS:INK 0,9:INK 1,26:BORDER 6       [1596]
360 ' +++ Grosse Ueberschrift           [2412]
370 LOCATE 1,25:PRINT "SLY SPRITE";     [1677]
380 x=232:y=392                         [978]
390 FOR hor=15 TO 2 STEP -2             [974]
400 y=y-4:x=232                         [1108]
410 FOR vert=1 TO 79                    [1501]
420 x=x+2                               [852]
430 pixel=TEST(vert,hor)                [2626]
440 IF pixel=0 THEN 470                 [1199]
450 MOVE x,y:DRAW x+2,y,1               [1070]
460 MOVE x,y-1:DRAW x+2,y-1             [1097]
470 NEXT                                [350]
480 NEXT                                [350]
490 LOCATE 10,2:PRINT "written in 1989" [2621]
500 LOCATE 56,2:PRINT "by Michael Lona" [2103]
510 LOCATE 1,25:PRINT STRING$(10," ")   [1626]
520 ' ++ Aufbau des Spielbretts         [1039]
530 FOR lx=40 TO 296 STEP 32            [1495]
540 MOVE 4,lx:DRAW 260,lx,1             [642]
550 NEXT                                [350]
560 FOR ly=4 TO 288 STEP 32             [1161]
570 MOVE ly,40:DRAW ly,296              [1731]
580 MOVE ly+1,40:DRAW ly+1,296          [1113]
590 NEXT                                [350]
600 FOR rand=261 TO 281 STEP 2          [1616]
610 MOVE rand,40+rand-261:DRAW rand,296+ra [2668]
nd-261
620 NEXT                                [350]
630 FOR rand=296 TO 316 STEP 4          [2344]
640 MOVE 4+rand-296,rand:DRAW 259+rand-296 [3028]
,rand
650 NEXT                                [350]
660 ' ++ Aufbau des Info-Windows        [1418]
670 LOCATE 48,5:PRINT STRING$(31,218)   [2067]
680 FOR rahmen=6 TO 22                  [1019]
690 LOCATE 48,rahmen:PRINT CHR$(207):LOCAT [5482]
E 78,rahmen:PRINT CHR$(207)
700 NEXT                                [350]
```

Listing SLY

```
710 LOCATE 48,23:PRINT STRING$(31,216)  [2703]
720 LOCATE 57,7:PRINT CHR$(24);" PUNKTESTA [2978]
ND ";CHR$(24)
730 LOCATE 50,10:PRINT name$(1)+":"     [2248]
740 LOCATE 50,12:PRINT name$(2)+":"     [1804]
750 LOCATE 54,20:PRINT "EINEN MOMENT BITTE [1978]
 !"
760 '                                   [117]
770 ' *** Zahlen verteilen ***          [1892]
780 '                                   [117]
790 ' ++ Leere Felder mit Pruefwert 100 be [1991]
legen
800 FOR wa=1 TO 8                       [797]
810 FOR se=1 TO 8                       [1178]
820 feld(wa,se)=100                     [1163]
830 NEXT                                [350]
840 NEXT                                [350]
850 ' ++ Zahlen per RND zuweisen        [1103]
860 FOR ausw=1 TO 64                    [351]
870 IF ausw<4 THEN sg=1 ELSE sg=0:' Keine [5206]
Zahl >10 in Startreihe (s. DATA - Zeilen)
880 READ zahl                           [1020]
890 RANDOMIZE TIME                      [1777]
900 wa=INT(RND*(8-sg))+(1+sg):se=INT(RND*8 [3815]
)+1
910 ' ++ Pruefen, ob Feld(wa,se) schon bel [3671]
egt ist
920 IF feld(wa,se)=100 THEN feld(wa,se)=za [4345]
hl ELSE 890
930 NEXT                                [350]
940 LOCATE 54,20:PRINT STRING$(20," ")  [2136]
950 ' ++ Zahlen in Spielfeld PRINTen    [1693]
960 xr=-2:yr=6                          [1012]
970 FOR se=1 TO 8                       [1178]
980 yr=yr+2                             [1073]
990 FOR wa=1 TO 8                       [797]
1000 xr=xr+4:IF xr=34 THEN xr=2         [2486]
1010 LOCATE xr,yr:PRINT USING "###";feld(w [1916]
a,se)
1020 NEXT                               [350]
1030 NEXT                               [350]
1040 ' ++ XOR-Grafikmodus an            [2012]
1050 PRINT CHR$(23);CHR$(1);            [1854]
1060 senky=287:senkx=8:horx=8:hory=287  [2416]
1070 ' ++ Cursor definieren             [1180]
1080 cu$=STRING$(3,CHR$(143))           [1242]
1090 GOSUB 1140                         [823]
1100 GOSUB 1310                         [948]
1110 '                                  [117]
1120 ' *** UP: Scores, Hinweise ***     [2231]
1130 '                                  [117]
1140 reihe=reihe+1:IF reihe=3 THEN reihe=1 [4046]
1150 LOCATE 70,10:PRINT USING "###";score( [2844]
1)
1160 LOCATE 70,12:PRINT USING "###";score( [1591]
2)
1170 LOCATE 50,15:PRINT "Letzter Zug:";lzu [3472]
g
1180 LOCATE 50,20:PRINT STRING$(13," ") [1521]
1190 LOCATE 50,20:PRINT name$(reihe)+"," [1010]
1200 LOCATE 50,21:PRINT "Sie sind an der R [3065]
eihe!"
1210 ' ++ Cursor nach Zug an letzte Positi [3478]
on setzen
1220 TAG                                [318]
1230 IF reihe=2 THEN 1240 ELSE 1260     [1981]
1240 MOVE senkx,senky:PRINT cu$;        [1603]
1250 GOTO 1270                          [357]
1260 MOVE horx,hory:PRINT cu$;          [1719]
1270 RETURN                             [555]
1280 '                                  [117]
1290 ' *** UP: Steuerung ***            [1867]
1300 '                                  [117]
1310 e$=INKEY$:IF e$="" THEN 1310       [936]
1320 IF INKEY(18)=0 OR JOY(0)=16 THEN 1520 [1087]
1330 IF (INKEY(47)=0 OR (JOY(0)>0 AND JOY( [3661]
0)<9)) AND reihe=1 THEN 1360
1340 IF (INKEY(47)=0 OR (JOY(0)>0 AND JOY( [4892]
0)<9)) AND reihe=2 THEN 1430 ELSE 1310
1350 ' ++ senkrechte Cursorbewegung     [1009]
1360 TAG                                [318]
1370 MOVE senkx,senky:PRINT cu$;        [1603]
1380 senky=senky-32                     [996]
1390 IF senky=31 THEN senky=287         [1456]
1400 MOVE senkx,senky:PRINT cu$;        [1603]
```

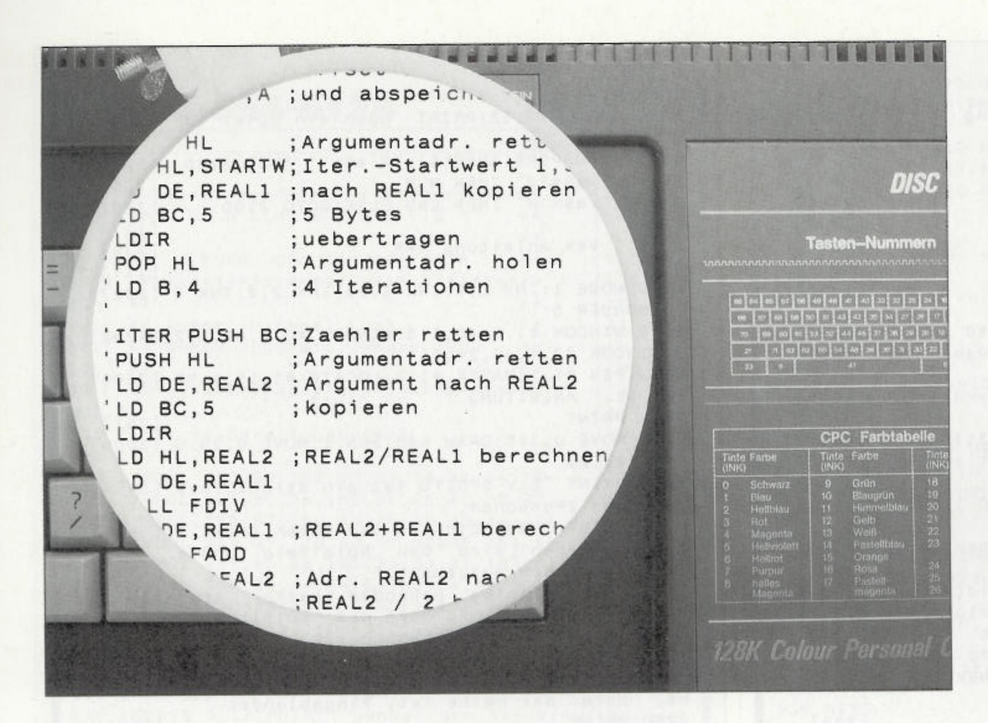
Listing SLY

```
1410 GOTO 1310                                    [436]
1420 ' ++ waagrechte Cursorbewegung               [1872]
1430 TAG                                          [318]
1440 MOVE horx,hory:PRINT cu$;                    [1719]
1450 horx=horx+32                                 [963]
1460 IF horx=264 THEN horx=8                      [2454]
1470 MOVE horx,hory:PRINT cu$;                    [1719]
1480 GOTO 1310                                    [436]
1490 '                                            [117]
1500 ' *** Zug ueberpruefen ***                   [1653]
1510 '                                            [117]
1520 IF reihe=1 THEN 1540 ELSE 1560               [2184]
1530 ' ++ Feld-Koordinaten der gewaehlten         [3787]
Zahl berechnen
1540 wa=(senkx+24)/32:se=9-((senky-31)/32)        [3085]
1550 GOTO 1580                                    [319]
1560 wa=(horx+24)/32:se=9-((hory-31)/32)          [2129]
1570 ' ++ Wenn Feld schon leer ist zurueck        [3815]
 zur Steuerungsabfrage
1580 IF feld(wa,se)=100 THEN SOUND 1,2000,        [2752]
30,15:GOTO 1310
1590 ' ++ Bestaetigung der korrekten Wahl         [2151]
1600 SOUND 1,200,45,5,,1                          [1831]
1610 score(reihe)=score(reihe)+feld(wa,se)        [3183]
1620 lzug=feld(wa,se):feld(wa,se)=100             [1632]
1630 ' ++ Pruefen ob Spiel beendet                [1343]
1640 IF reihe=1 THEN 1650 ELSE 1670               [2091]
1650 FOR l=1 TO 8:s=s+feld(l,se):NEXT:IF s        [1993]
=800 THEN 1800
1660 GOTO 1680                                    [395]
1670 FOR l=1 TO 8:s=s+feld(wa,l):NEXT:IF s        [1478]
=800 THEN 1800
1680 s=0                                          [285]
1690 TAGOFF                                       [1066]
1700 LOCATE 4*wa-2,2*se+6:PRINT "  "              [2445]
1710 IF reihe=1 THEN 1720 ELSE 1740               [1363]
1720 horx=senkx:hory=senky                        [2107]
1730 GOTO 1750                                    [349]
1740 senkx=horx:senky=hory                        [1993]
1750 GOSUB 1140                                   [823]
1760 GOTO 1310                                    [436]
1770 '                                            [117]
1780 ' *** Spielende ***                          [887]
1790 '                                            [117]
1800 MODE 1:INK 0,22:INK 1,2:INK 2,6:INK 3        [3265]
,25:BORDER 22
1810 ' ++ Gewinner ermitteln                      [1144]
1820 IF score(1)>score(2) THEN win=1:GOTO         [3954]
1850
1830 IF score(2)>score(1) THEN win=2:ELSE         [3019]
GOTO 2030
1840 ' ++ Urkunde aufbauen                        [2286]
1850 WINDOW 9,31,1,25:PAPER 3:CLS                 [2119]
1860 PEN 2                                        [548]
1870 LOCATE 1,1:PRINT CHR$(150);STRING$(21        [2342]
,CHR$(154));CHR$(156)
1880 LOCATE 1,25:PRINT CHR$(147);STRING$(2        [2988]
1,CHR$(154));CHR$(153);
1890 FOR umr=2 TO 24                              [1596]
1900 LOCATE 1,umr:PRINT CHR$(149):LOCATE 2        [4363]
3,umr:PRINT CHR$(149)
1910 NEXT                                         [350]
1920 LOCATE 9,3:PRINT urkunde$                    [1718]
1930 PEN 1:LOCATE 3,7:PRINT "Name: ";:PEN         [3540]
2:PRINT name$(win)
1940 PEN 1:LOCATE 3,9:PRINT "Punktzahl:";:        [5022]
PEN 2:PRINT score(win)
1950 PEN 1:LOCATE 3,14:PRINT "Dem Besitzer        [3955]
 dieser"
1960 LOCATE 9,15:PRINT "Urkunde"                  [1275]
1970 LOCATE 3,16:PRINT "wird der Titel des        [2790]
"
1980 PEN 2:LOCATE 4,18:PRINT "SLY SPRITE-M        [1997]
ASTERS"
1990 PEN 1:LOCATE 8,20:PRINT "verliehen"          [1691]
2000 LOCATE 3,24:PRINT "Die Spielleitung"         [2822]
2010 CALL &BB18:GOTO 2080                         [1265]
2020 ' ++ Unentschieden                           [978]
2030 PEN 3:LOCATE 13,12:PRINT "UNENTSCHIED        [1718]
EN!"
2040 CALL &BB18:GOTO 2080                         [1265]
2050 '                                            [117]
2060 ' *** Neues Spiel (J/N)? ***                 [2095]
2070 '                                            [117]
```

Listing SLY

```
2080 MODE 1:PEN 2:PAPER 0                         [1915]
2090 LOCATE 9,12:PRINT "Noch ein Spiel (J/        [3060]
N) ?"
2100 e$=UPPER$(INKEY$):IF e$="" THEN 2100         [2226]
2110 IF e$="J" THEN RUN                           [706]
2120 IF e$="N" THEN END:ELSE GOTO 2100            [726]
2130 '                                            [117]
2140 ' *** Anleitung ***                          [939]
2150 '                                            [117]
2160 MODE 1:INK 0,0:INK 1,26:INK 2,2:INK 3        [2327]
,6:BORDER 0
2170 WINDOW #1,1,40,1,3:WINDOW #0,1,40,4,2        [4788]
3:WINDOW #2,1,40,24,25
2180 PEN #1,1:PAPER #1,3:LOCATE #1,15,1:PR        [2596]
INT #1," ANLEITUNG "
2190 PRINT                                        [361]
2200 MOVE 0,355:DRAW 640,355,1:MOVE 0,29:D        [3009]
RAW 640,29
2210 PRINT "SLY SPRITE ist ein Strategiesp        [5446]
iel fuer 2Personen."
2220 PRINT "Nach Eingabe der Namen (max. 1        [9490]
3 Buchsta-ben) wird das  Spielfeld  und e
in Info-Window aufgebaut."
2230 PRINT "Einige  Sekunden spaeter  ersc        [10155]
heinen  inden 64  Feldern des  Spielbretts
  Zahlenmit Werten zwischen +20 und -15."
2240 PRINT "Im Info-Window sind die Scores        [12618]
,der letz-te Zug, und der  Name des Spiele
rs,  deran der Reihe ist, eingeblendet."
2250 PRINT                                        [361]
2260 PRINT "Der Spieler, der am Zug ist, s        [17788]
ucht  sichmit  der  SPACE-Taste  oder  IRG
ENDEINERJoystickbewegung  die  gewuenschte
  Zahlaus  und  bestaetigt  mit  <ENTER>
oder<FIRE>. Die genommene Zahl wird  zu se
i-nem Punktestand hinzuaddiert."
2270 GOSUB 2360                                   [819]
2280 PRINT "Dabei ist zu beachten, dass di        [13762]
e  Auswahldes ersten  Spielers auf  die se
nkrechteReihe, und die des zweiten  Spiele
rs aufdie waagrechte Reihe beschraenkt ist
."
2290 PRINT "Hat ein Spieler eine Zahl geno        [22046]
mmen, musssein Gegner in DER Reihe (senkre
cht bzw.waagrecht)  weitermachen,  in der
 dieseZahl war. Vor allem im fortgeschritt
enenSpielverlauf kann somit der Gegenspiel
erzu bestimmten Zuegen gezwungen werden."
2300 PRINT                                        [361]
2310 PRINT "Nach Spielende  wird das  Erge        [7588]
bnis ange-zeigt. Druecken  Sie danach  bit
te  eineTaste."
2320 PEN 2:LOCATE 5,19:PRINT "UND NUN VIEL        [3270]
 SPASS MIT SLY SPRITE"
2330 GOSUB 2360                                   [819]
2340 PEN 1:GOTO 270                               [1139]
2350 ' ++ UP: Textseite "langsam" loeschen        [1619]
2360 PEN #2,3:LOCATE #2,14,2:PRINT #2,"TAS        [3040]
TE DRUECKEN";
2370 CALL &BB18                                   [389]
2380 LOCATE #2,14,2:PRINT #2,STRING$(14,"         [2108]
");
2390 PRINT CHR$(23);CHR$(1);                      [1854]
2400 FOR a=47 TO 195 STEP 4                       [1450]
2410 SOUND 1,100+a,2                              [1197]
2420 MOVE 0,a:DRAW 640,a,2                        [1247]
2430 MOVE 0,386-a:DRAW 640,386-a                  [2028]
2440 NEXT a                                       [383]
2450 PRINT CHR$(23);"0";                          [1485]
2460 FOR a=1 TO 149 STEP 2                        [1582]
2470 SOUND 1,295-a,2                              [835]
2480 MOVE 0,195-a:DRAW 640,195-a,0                [825]
2490 MOVE 0,195+a:DRAW 640,195+a                  [2574]
2500 NEXT a                                       [383]
2510 CLS:PRINT                                    [356]
2520 RETURN                                       [555]
2530 ' ++ DATAs fuer die Zahlen auf dem Sp        [3219]
ielbrett
2540 DATA 20,15,12,10,10,9,9,8,8,7,7,6,6,6        [4013]
,5,5,5,4,4,4,4,3,3,3,3,2,2,2,2,1,1,1,1,1,0
,0,0
2550 DATA 0,0,0,-1,-1,-1,-1,-1,-2,-2,-2,-2        [4651]
,-2,-3,-3,-3,-4,-4,-4,-5,-5,-6,-6,-7,-8,-1
0,-15
```

Listing SLY



Die Assemblerecke

Maschinensprache mit Taktgefühl

Wenn sich Basic, Pascal, C und Konsorten für eine Problemstellung als zu lahm erweisen, bleibt immer noch der Ausweg, den Rechner in Maschinensprache auf Trab zu bringen. Was aber tun, wenn eine Assemblerroutine zu langsam arbeitet? Echtzeit-Datenerfassung, animierte Grafik oder Fraktal-Berechnungen können selbst die schnellsten Prozessoren überfordern. Doch angesichts solcher Probleme kommt der Assembler-Spezialist erst richtig in Form und beginnt, in seiner Trickkiste zu wühlen...

Eines der wichtigsten Werkzeuge bei der Optimierung von Assemblerprogrammen ist eine Tabelle, die angibt, wie viele Taktzyklen der Prozessor für die verschiedenen Maschinenbefehle benötigt. Der Z80 wird im CPC mit einer Frequenz von 4 MHz (Megahertz, Millionen Schwingungen pro Sekunde) betrieben. Jeder Taktimpuls, der irgendwelche elektronischen Umschaltvorgänge auslöst, dauert demnach 0,25 Mikrosekunden. Der Tabelle der Z80-Ausführungszeiten ist zu entnehmen, daß die Ausführung eines Befehls zwischen 4 und 23 Takten und damit mindestens eine Mikrosekunde beansprucht.

Leider kommt im CPC noch ein bremsender Faktor hinzu. Sowohl der Prozessor als auch die Video-Logik greifen auf den RAM-Speicher zu und sind deshalb gezwungen, ihre Aktivitäten zu koordinieren. Diese Koordination erfolgt im Mikrosekunden-Raster, so daß der Z80 mitunter etwas warten muß, bis er mit dem nächsten Befehl beginnen darf. Das Kommando POP AX benötigt beispielsweise zehn Takte bzw. 2,5 Mikrosekunden. Danach sind erst einmal zwei Wartezyklen bis zur nächsten vollen Mikrosekunde fällig. Oder mit anderen Worten: Für den CPC muß die vom Z80-Hersteller Zilog angegebene Anzahl Taktzyklen auf den nächsten durch 4 teilbaren Wert aufgerundet werden!

Zeitanteile

Der Zeitbedarf eines Befehls setzt sich aus zwei Komponenten zusammen:

- Lesen der Opcode-Bytes aus dem Speicher und Dekodierung
- Ausführung des Befehls

Der erste Anteil ist um so größer, je mehr Bytes der Befehlscode umfaßt. INC HL (1 Byte Opcode) braucht zum Beispiel sechs Takte; INC IX (2 Byte Opcode) insgesamt zehn, obwohl beide Befehle im Prinzip das gleiche erledigen. Der zweite Anteil hängt davon ab, wie komplex der Befehl ist. Besonders Speicherzugriffe kosten Zeit, da über den Adreßbus erst die richtige RAM- oder ROM-Speicherzelle angesprochen werden muß. Aus diesem Grund braucht LD A,B nur vier Takte, LD A,(HL) dagegen sieben Takte, obwohl beide Befehle nur ein Opcode-Byte belegen. Recht 'teuer' ist LD A,(IX+Ofs): 3 Bytes Opcode, ein Speicherzugriff und eine arithmetische Operation für die Berechnung der absoluten Adresse ergeben 19 Takte.

Aus diesen Erkenntnissen lassen sich zwei Faustregeln für die Optimierung von Assemblerprogramme ableiten:

1) Benutzen Sie möglichst kurze Befehle. Wenn Sie den Akku mit SUB A löschen, kommen Sie mit einem Byte und vier Takten aus. Die 2-Byte-Alternative LD A,0 (7 Takte) ist nur angebracht, wenn die Flags unbeeinflußt bleiben sollen.

2) Vermeiden Sie unnötige Speicherzugriffe. Halten Sie Variablen und Zwischenergebnisse möglichst in den Prozessorregistern.

Falls die Register schon alle belegt sind, sollten Sie fleißig den Z80-Stack benutzen. PUSH und POP ermöglichen einen sehr schnellen Speicherzugriff: Mit nur zwölf Taktzyklen werden 2 Bytes transportiert, das schafft kein anderer Befehl! Angenommen, ein 500 Byte langer Bereich ab Adresse &A000 soll mit Nullen gefüllt werden. Normalerweise benutzt man den Blocktransferbefehl LDIR:

```
LD BC,499       ;Anzahl Bytes -
1 LD HL,&A000   ;Startadresse
LD DE,&A001     ;Startadresse + 1
LD (HL),0       ;1. Byte vorbelegen
LDIR            ;Bereich füllen
```

Die Schleifenautomatik des LDIR-Kommandos kopiert im ersten Durchlauf die Null von &A000 nach &A001, im zweiten Durchlauf von &A001 nach &A002, usw. Diese Methode ist relativ schnell, benötigt allerdings pro Kopiervorgang einen Lese- und einen Schreibzugriff auf den Speicher. Mit den Tabellenwerten (ohne Berücksichtigung der CPC-Wartezyklen) ergibt sich eine Summe von 10519 Takten. Erstaunlicherweise läßt sich der Zeitaufwand auf 5821 Taktzyklen reduzieren, indem man trickreich den Befehl PUSH verwendet:

```
DI                ;Interrupts sperren
LD (SPSAVE),SP    ;SP retten
LD SP,&A1F4       ;SP = &A000+500
LD B,250          ;Zähler = 500/2
LD HL,0           ;HL = 0
LOOP PUSH HL      ;in den Speicher
```

Möchten Sie gerne programmieren ?

CPC-BASIC-Lehrbuch



Nutzen Sie die vollen Möglichkeiten der leistungsstarken CPC-Hardware durch das spezielle Schneider-BASIC!
Um alle hardwaremäßigen Möglichkeiten der CPC-Computer nutzen zu können, wurde von Amstrad/Schneider extra für diesen Computertyp ein eigener BASIC-Dialekt entwickelt: Das Schneider-CPC-Basic. Nur mit diesen speziellen Befehlen und Optionen können Sie den Leistungsumfang Ihres CPC voll ausschöpfen.

Das Lehrbuch

Auf 150 Seiten wird Ihnen im handlichen Ringordner nicht nur der komplette Befehlsumfang des Schneider-BASIC dokumentiert, mit dem Index und einer Hervorhebung der Schlüsselwörter können Sie auch bequem nachschlagen und Ihre Kenntnisse auffrischen.
Neben den Erklärungen der Schneider-BASIC-Befehle in gedruckter Form vermittelt Ihnen das Lehrbuch durch Beschreibung die korrekte Verwendung. Mit Beispielen und Listings zur praktischen Arbeit am Computer wird danach der Einsatz in der Programmierung demonstriert.
Die zwölf Kapitel des Lehrbuchs behandeln den gesamten Stoff systematisch und nach lerntechnischen Gesichtspunkten in Form eines Kurses. Durch Fragen am Ende jedes Kapitels können Sie Ihren Lernfortschritt überprüfen und kontrollieren.

Die Software

Auf einer Kassette erhalten Sie alle Programme und Listings des Lehrbuchs, um die beschriebenen Techniken nachvollziehen zu können. Dazu noch eine Reihe von Spielen zu Ihrem Vergnügen und Programme, die die Möglichkeiten Ihres CPC demonstrieren.
Die zweite Kassette enthält die Lernziel-Kontrolltests für jedes der zwölf Kapitel: Sie können dadurch direkt im Dialog mit dem Computer Ihre Lernerfolge kontrollieren.



Lehrbuch mit Kassette

DM 29,– *

CPC-Assembler-Software und Trainingsbuch

Das Buch

führt den Anfänger schrittweise in die Programmierung des Z80 ein. Dabei werden Vorkenntnisse nicht vorausgesetzt. Nach der Lektüre des Buchs sind Sie mit dem Befehlssatz des Prozessors wie auch mit den Adressierungsarten vertraut. Anhand einer Fülle von Programmbeispielen, die speziell auf den CPC 464 zugeschnitten sind, lernen Sie, nützliche Routinen in Maschinensprache zu entwickeln, die auf Ihrem Rechner sofort lauffähig sind. Dazu erfahren Sie, wie Sie die im ROM des Schneider CPC vorhandenen Hilfsroutinen sinnvoll einsetzen können.
Darüber hinaus lernen Sie Programme zur Erweiterung des Betriebssystems mit leistungsfähigen, grafischen Befehlen kennen, die Sie in BASIC verwenden können, z.B. TRIANGEL, BOX und CIRCLE. Ein umfangreicher Anhang mit Erläuterungen des Assembler-Befehlssatzes sowie einer Vielzahl kommentierter Einsprungadressen des Betriebssystem-ROMs rundet das Buch ab.

Die Software

besteht aus einem menügesteuerten Z80 - Assembler. Mit seiner Hilfe können Sie nicht nur die im Buch erläuterten Beispielprogramme editieren und in Maschinensprache übersetzen, sondern auch selbstentworfene Programme entwickeln und testen. Für Ihre Arbeit stehen Ihnen Funktionen zur Verfügung wie **Einfügen – Löschen – Ändern von Programmzeilen – Abspeichern und Laden von Programmen auf Datenträgern wie Kassette und Diskette – Ausgabe von Listings auf Bildschirm und Drucker – Zahlenkonvertierung von der Menü-Ebene aus – Durchführung von Zwischenrechnungen mit binären, hexadezimalen und dezimalen Zahlen.** Dazu erhalten Sie ein Trainingsprogramm, mit dessen Hilfe Sie den Umgang mit verschiedenen Zahlenformaten vertiefen können, und Programme zur Demonstration der grafischen Erweiterungen des Betriebssystems.



Buch mit Kassette

39,– DM*

Buch mit Diskette

49,– DM*

* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung.

Bitte benutzen Sie die Bestellkarte.





```
DJNZ LOOP      ;-> 250 mal
LD SP,(SPSAVE) ;alter SP zurück
EI             ;Interrupts erlauben
```

Das SP-Register (Stackpointer) wird hier mit der Endadresse des Speicherbereichs geladen, da der Stack von oben nach unten wächst. PUSH in Verbindung mit dem Schleifenbefehl DJNZ stapelt dann 250 Doppelnullen. Natürlich muß der Stackpointer zu Beginn irgendwo abgespeichert und zum Schluß wiederhergestellt werden, damit die Stapelverwaltung des Z80 nicht durcheinander gerät. DI unterbindet den CPC-Systeminterrupt. Da die Interruptroutine mit diversen CALLs, PUSHs und POPs den Z80-Stack benutzt, würde sie sonst unkontrolliert in den Speicherbereich schreiben.

Doppelt hält besser

Ein anderes Mittel zur Vermeidung von Speicherzugriffen ist die Verwendung des Zweitregistersatzes. Die Prozessorregister AF, BC, DE und HL sind im Z80 doppelt vorhanden und werden zur Unterscheidung von den 'normalen' Registern mit einem Apostroph gekennzeichnet (zum Beispiel AF'). Für den Zugriff auf die Zweitregister stehen zwei Befehle zur Verfügung: EX AF,AF' tauscht den Akku und die Flags des Erstregistersatzes gegen die entsprechenden Zweitregister aus. Danach arbeitet der Prozessor mit den 'alternativen' Registern, bis ein weiterer Austausch erfolgt. EXX tauscht analog die restlichen Register (BC, DE, HL) aus. Beide Befehle werden sehr schnell abgearbeitet (vier Takte), da sie keine Daten bewegen. Sie beeinflussen nur einen internen Schalter, der bestimmt, welcher der beiden Registersätze aktiv ist.
Damit taucht die Frage auf, warum wir diese nützliche Einrichtung in den bisherigen Folgen der Assemblerecke so sträflich vernachlässigt haben. Dafür gibt es jedoch (leider) einen guten Grund: Der Zweitregistersatz wird im CPC bereits vom Betriebssystem mit Beschlag belegt, um wichtige Informationen aufzubewahren:
– Während des Systeminterrupts wird der Zweitregistersatz eingeschaltet und auch fleißig benutzt, um den Inhalt der Erstregister zu schonen. Die unterbrochene Routine darf ja in keiner Weise in ihrer Arbeit beeinträchtigt werden.
– In F' zeigt das Carryflag an, ob sich der CPC gerade in der Interruptbehandlung befindet. Dadurch kann die Interruptroutine erkennen, ob sie sich selbst unterbrochen hat, was spezielle Maßnahmen erfordert.
– In B' steht die Portadresse des Gate Arrays (&F7) und in C' der aktuelle Inhalt des Gate-Array-Registers R2, das unter anderem für den RAM/ROM-Status zuständig ist (siehe Heft 12/89). Bei jedem Aufruf einer ROM-Routine über die Restart-Vektoren schaltet das Betriebssystem kurzfristig den Zweitregistersatz ein, um dann mit Hilfe des Befehls OUT (C),C den RAM/ROM-Status umzuschalten (siehe letzte Folge).
Die Geschwindigkeitsvorteile, die sich aus der Verwendung des Zweitregistersatzes ergeben, werden also schon weitgehend vom CPC-Betriebssystem 'verbraten'. Will man trotzdem auf die Reserveregister zugreifen, so sollte zu Beginn der Routine der Systeminterrupt unterbunden werden, da er wie gesagt die Zweitregister überschreibt. Die Register F' und BC' müssen gerettet und zum Abschluß wieder hergestellt werden, damit die nächste Umschaltung der Speicherkonfiguration nicht im Chaos endet. Der Aufruf von Betriebssystemroutinen über die Restart-Vektoren (zum Beispiel das übliche CALL &BB5A für die Textausgabe) ist strikt zu vermeiden! Wer mit diesen Einschränkungen leben kann, sollte sich an dem folgenden 'Rahmenprogramm' orientieren:

```
DI          ;Interrupts unterbinden
EX AF,AF'   ;Zweitregister ein EXX
PUSH AF     ;Interrupt-Flag und
PUSH BC     ;Sytemstatus retten
;Hier koennen Sie mit Erst-
;und Zweitregistern arbeiten
POP BC      ;Systemstatus
POP AF      ;und Interrupt-Flag
EXX         ;wieder in den
EX AF,AF'   ;Zweitregistersatz
EI          ;Interrupts erlauben
```

Die erhöhte Anzahl Prozessorregister kann in einigen Fällen eine Menge Rechenzeit ersparen. Der Vorteil ist jedoch oft nicht so groß, wie man auf den ersten Blick vermutet. Das liegt insbesondere an den fehlenden Direktverbindungen zwischen Erst-und Zweitregistersatz. Um einen 16-Bit-Wert von HL nach HL' zu kopieren, sind zum Beispiel mindestens drei Befehle notwendig (PUSH HL, EXX, POP HL).

Unterbrechungen unerwünscht

Übrigens bringt schon alleine das Ausschalten des Systeminterrupts eine Geschwindigkeitssteigerung, da dieser den normalen Programmablauf 300mal pro Sekunde unterbricht und damit etwa 9 Prozent der Rechenzeit beansprucht. Wer sein gesamtes Programm auf diese Weise beschleunigen will, sollte jedoch folgendes bedenken:
– Alle interruptgesteuerten Betriebssystem-Aktivitäten des CPC (Systemuhr, Tastaturabfrage, Farbwechsel, Tonerzeugung..) werden lahmgelegt.
– Jeder Aufruf einer Betriebssystemroutine über die Restart-Vektoren (also zum Beispiel CALL &BB5A) schaltet die Interrupts wieder ein.
Der Grund: Die Restart-Routinen greifen beim Umschalten der Speicherkonfiguration auf den Zweitregistersatz zu und müssen diese Passage durch ein DI/EI-Paar vor Unterbrechungen schützen. Der letzte Befehl ist immer EI (Enable Interrupt). Es gibt jedoch noch eine andere Möglichkeit, den Systeminterrupt zu eliminieren, ohne daß ein EI daran etwas ändern kann. Wie in der letzten Folge beschrieben, führt ein Interruptsignal zum Aufruf einer Routine, die an der Speicherstelle &0038 beginnt (RST &38). Beim CPC steht dort ein JP-Befehl, der zu der eigentlichen Interruptroutine springt. Ersetzt man den Sprung durch ein RET-Kommando, endet jeder Interrupt mit einem sofortigen Rücksprung zur unterbrochenen Routine. In der Praxis sieht die Aktion so aus:

```
LD A,&C9     ;Opcode für RET
LD (&38),A   ;Interrupt unterbinden
;dazwischen der Programmcode
LD A,&C3     ;Opcode für JP
LD (&38),A   ;Interrupt ein
```

Zu beachten ist dabei, daß der Sprungbefehl an der Speicherstelle &0038 im CPC doppelt vorhanden ist: Einmal im RAM und einmal an gleicher Stelle im unteren ROM, da die Interruptbehandlung ja bei jeder Speicherkonfiguration funktionieren muß. Solange im unteren Adreßbereich (0000..&3FFF) das ROM selektiert ist (zum Beispiel während der Ausführung einer Betriebssystem-Routine), bleibt der obige Sabotageakt also unwirksam!

Selbst-Modifikationen

Zum Abschluß soll noch eine weitere trickreiche Methode vorgestellt werden, um Assemblerprogramme zu beschleunigen. Angenommen, in einer Routine wird ein 8-Bit-Wert im B-Register benötigt, der vorher an anderer Stelle im A-Register berechnet wurde. Wenn keine Möglichkeit besteht, das Ergebnis in den Prozessorregistern oder auf dem Stack aufzubewahren, bleibt nur die Auslagerung in eine

Speicherstelle (hier mit dem Label ERGEB versehen):

```
LD (ERGEB),A  ;Ergebnis abspeichern
; dazwischen weiterer Programmcode
LD A,(ERGEB)  ;Ergebnis holen
LD B,A        ;und nach B kopieren
```

Kürzer und schneller ist eine Lösung, die den Aufbau der Z80-Opcodes ausnutzt. Es ist ja ohne weiteres möglich, einen konstanten Wert unmittelbar in das B-Register zu laden (zum Beispiel LD B,10). Das erste Opcode-Byte lautet in diesem Fall &06, das zweite Byte enthält den Wert. Diese Tatsache lädt dazu ein, das Ergebnis direkt in den Programmcode einzusetzen. Da sich der vollständige Befehl erst zur Laufzeit des Programms herausstellt, müssen wir mit Assemblerdirektiven arbeiten. DB definiert das erste Opcode-Byte; der Platz für das zweite Byte wird durch DS (Define Space) freigehalten:

```
LD (ERGEB),A  ;Ergebnis abspeichern
;dazwischen weiterer Programmcode
DB &06        ;Opcode für LD B,Wert
ERGEB DS 1    ;1 Byte freilassen
```

Das Verfahren spart hier einen Speicherzugriff und das Umkopieren von A nach B. Unter Berücksichtigung der CPC-Wartezyklen ergibt sich ein Vorteil von zwölf Takten! Ein selbstmodifizierender Code könnte auch 16-Bit-Werte oder Sprungadressen im Programm abändern. Ein eventueller Nachteil liegt darin, daß der Code aus naheliegenden Gründen nicht in einem ROM oder EPROM funktioniert. Weiterhin gelten selbstmodifizierende Programme unter Profis als 'unfein', da sie unübersichtlich und fehleranfällig sind. Den Trick sollte man also nur anwenden, wenn zwingende Gründe vorliegen.

Beim CPC ist er mitunter angebracht, wenn man eine Betriebssystem-Routine direkt im ROM aufrufen will, ohne sich Ärger mit den verschiedenen CPC-Versionen einzuhandeln. Auf diese Weise kann man den Zeitverlust verhindern, der bei jedem CALL durch die Abarbeitung der in der letzten Folge beschriebenen RST-&08-Routine entsteht. Die Zeitersparnis wirkt sich besonders bei kurzen Betriebssystem-Routinen aus, bei denen die Umschaltung der Speicherkonfiguration einen hohen prozentualen Anteil an den gesamten 'Betriebskosten' hat. Ein lohnendes Beispielobjekt ist die Routine SCREEN NEXT LINE, da sie häufig in Schleifen Verwendung findet. Sie wird normalerweise über CALL &BC26 aufgerufen und dient dazu, in Grafikanwendungen eine Bildschirmadresse in HL eine Rasterzeile nach unten weiterzurechnen.

Hier nun in groben Zügen die praktische Realisierung: Am Anfang des Programms steht ein RST-&28-Sprungbefehl (FIRM JUMP), der wie ein JP-Befehl wirkt und nur den Zweck hat, für den gesamten Programmablauf das untere ROM einzuschalten. Damit das Programm auf allen CPCs funktioniert, wird die ROM-Adresse aus dem Sprungvektor für SCREEN NEXT LINE gelesen und 'selbstmodifizierend' in den Programmcode geschrieben. Vorher muß das Programm allerdings noch die beiden höchstwertigen Bits zurücksetzen, da sie nicht zur Adresse gehören, sondern Angaben zur Speicherkonfiguration enthalten (siehe letzte Folge).

```
RST &28              ;unteres ROM einschalten
DW START             ;durch RST-&28-Sprung
START LD HL,(&BC27)  ;ROM-Adr. lesen
RES 6,H              ;die beiden Konfigura-
RES 7,H              ;tionsbits löschen
LD (ROMADR),HL       ;ROM-Adr. einsetzen
;...und dann irgendwo im Programm:
DB &CD               ;Opcode für CALL
ROMADR DS 2          ;Platz für ROM-Adresse
```

Und damit genug für dieses Mal. Die nächste Folge der Assemblerecke wird zeigen, wie Sie durch tabellenorientierte Arithmetik weitere kostbare Mikrosekunden einsparen können.

(Matthias Uphoff/cd)

Z80 Taktzyklen

Befehl	Operand	Takte	Befehl	Operand	Takte	Befehl	Operand	Takte
LD	rg,rg	4	SBC	A,rg	4	SRA	rg	8
LD	rg,Wert	7	SBC	A,Wert	7	SRA	(HL)	15
LD	rg,(HL)	7	SBC	A,(HL)	7	SRA	(ir+Ofs)	23
LD	rg,(ir+Ofs)	19	SBC	A,(ir+Ofs)	19			
LD	A,(BC)	7				SRL	rg	8
LD	A,(DE)	7	CP	rg	4	SRL	(HL)	15
LD	A,(Adr)	13	CP	Wert	7	SRL	(ir+Ofs)	23
			CP	(HL)	7			
LD	(HL),rg	7	CP	(ir+Ofs)	19	RLD		18
LD	(HL),Wert	10				RRD		18
LD	(ir+Ofs),rg	19	AND	rg	4			
LD	(BC),A	7	AND	Wert	7	BIT	b,rg	8
LD	(DE),A	7	AND	(HL)	7	BIT	b,(HL)	12
LD	(Adr),A	13	AND	(ir+Ofs)	19	BIT	b,(ir+Ofs)	20
LD	A,I	9	OR	rg	4	SET	b,rg	8
LD	A,R	9	OR	Wert	7	SET	b,(HL)	15
LD	I,A	9	OR	(HL)	7	SET	b,(ir+Ofs)	23
LD	R,A	9	OR	(ir+Ofs)	19			
						RES	b,rg	8
LD	dr,Wert	10	XOR	rg	4	RES	b,(HL)	15
LD	ir,Wert	14	XOR	Wert	7	RES	b,(ir+Ofs)	23
LD	HL,(Adr)	16	XOR	(HL)	7			
LD	dr,(Adr)	20	XOR	(ir+Ofs)	19	JP	Adr	10
LD	ir,(Adr)	20				JP	Bed,Adr	10
			ADD	HL,dr	11	JP	(HL)	4
LD	(Adr),HL	16	ADC	HL,dr	15	JP	(ir)	8
LD	(Adr),dr	20	SBC	HL,dr	15	JR	Ofs	12
LD	(Adr),ir	20	ADD	ir,dr	15	JR	Bed,Ofs	7/12**
LD	SP,HL	6				DJNZ	Ofs	8/12**
LD	SP,ir	10	INC	rg	4			
			INC	(HL)	11	CALL	Adr	17
PUSH	dr	11	INC	(ir+Ofs)	23	CALL	Bed,Adr	10/17**
PUSH	ir	15	INC	dr	6	RET		10
POP	dr	10	INC	ir	10	RET	Bed	5/10**
POP	ir	14						
			DEC	rg	4	RST	Adr	11
EX	DE,HL	4	DEC	(HL)	11			
EX	AF,AF'	4	DEC	(ir+Ofs)	23	DAA		4
EXX		4	DEC	dr	6	CPL		4
EX	(SP),HL	19	DEC	ir	10	NEG		8
EX	(SP),ir	23				SCF		4
			RLCA		4	CCF		4
LDI		16	RLA		4	NOP		4
LDIR		n x 21*	RRCA		4	DI		4
LDD		16	RRA		4	EI		4
LDDR		n x 21*						
CPI		16	RLC	rg	8	IN	rg,(C)	12
CPIR		n x 21*	RLC	(HL)	15	IN	A,(Wert)	11
CPD		16	RLC	(ir+Ofs)	23	OUT	(C),rg	12
CPDR		n x 21*				OUT	(Wert),A	11
			RL	rg	8			
ADD	A,rg	4	RL	(HL)	15			
ADD	A,Wert	7	RL	(ir+Ofs)	23	rg	8-Bit-Register	
ADD	A,(HL)	7					A,B,C,D,E,H,L	
ADD	A,(ir+Ofs)	19	RRC	rg	8	dr	Doppelregister	
			RRC	(HL)	15		BC,DE,HL oder SP	
ADC	A,rg	4	RRC	(ir+Ofs)	23	ir	Indexregister	
ADC	A,Wert	7					IX,IY	
ADC	A,(HL)	7	RR	rg	8	Bed	Bedingung C,NC,	
ADC	A,(ir+Ofs)	19	RR	(HL)	15		Z,NZ,P,M,PO,PE	
			RR	(ir+Ofs)	23			
SUB	rg	4				*	Ausführungszeit pro	
SUB	Wert	7	SLA	rg	8		Wiederholung	
SUB	(HL)	7	SLA	(HL)	15	**	2. Wert wenn Sprung	
SUB	(ir+Ofs)	19	SLA	(ir+Ofs)	23		andernfalls 1. Wert	

Tabelle 1: Die Assemblerbefehle und ihre Ausführzeiten (Taktzyklen)



100,– DM für 1 kByte

Die Herausforderung

Blättert man in der Weltliteratur an diesen und jenen Stellen, so bemerkt man schnell, daß sich die klügsten Köpfe mit dem Thema "Kurz und Knapp" herumgeschlagen haben. Nietzsche, Sophokles, Shakespeare, – alle konnten sich nicht zurückhalten, ihre Kommentare zu diesem Thema abzugeben. "Alles Schwätzer", werden viele sagen, "der Worte sind genug..., etc.". Die so verlangten "Taten der Kürze" werden von unseren Lesern Monat für Monat in fantastischen Kleinstprogrammen abgeliefert. Lehnen wir uns also wieder einmal zurück und lassen die sechs Februarwerke auf uns wirken.

MULTICOLOR

Das Problem der geringen Farbwahl im Mode 1 des Amstrad CPCs taucht in derart verschiedenen Variationen auf, daß es nur logisch erscheint, wenn dem eine Vielzahl von Lösungswegen entgegengesetzt wird.

Viele CPC-Spiele benutzen den Grafikmodus 'MODE 0'. So können zwar 16 Farben gleichzeitig auf dem Bildschirm dargestellt werden, die reale Grafikauflösung ist aber sehr niedrig (160*200). Grafikmodus 1 hingegen liefert ein wesentlich schärferes Bild, es sind aber nur vier Farben gleichzeitig darstellbar.

Das Programm 'Multicolor' ermöglicht durch einen etwas umständlich erscheinenden Trick die **gleichzeitige** Darstellung von sechs Vordergrundfarben und einer Hintergrundfarbe in MODE 1. Dadurch kann sicher in einigen Fällen auf die bessere Grafik von MODE 1 zurückgegriffen werden.

Das Programm ist nicht relokativ.

Aufruf von BASIC:

'CALL &A000,1' – neue Farbroutine wird aktiviert.

'CALL &A000,0' – Die Betriebssystem-Routine wird wiederhergestellt.

Im Schneider BASIC können beim INK-Befehl außer dem Farbstift zwei Farbwerte übergeben werden, die dann gegeneinander blinken. Im Speicher existieren zwei Farbtabellen, in denen die Farbwerte der (eventuell blinkenden) PENs gespeichert werden. Das Programm 'Multicolor' stellt im obersten Fünftel des Bildschirms die PENs 1 bis 3 mit den Farben der ersten Farbtabelle, den Rest des Bildschirms mit den Farben der zweiten Farbtabelle dar.

Das Blinken der Farben wird also **nicht** mehr ausgeführt, sondern in den obersten fünf Zeilen sind zum Beispiel die PENs 1 bis 3 blau, rot und gelb, in den unteren 20 Zeilen jedoch zum Beispiel purpur, grün und weiß. Hintergrund- und Rahmenfarbe bleiben unverändert in beiden 'Bildschirmteilen' gleich, blinken jedoch ebenfalls nicht. Will der BASIC-Benutzer nun also mehr Farben in MODE 1 darstellen, muß er nur entsprechende INK-Befehle geben, die nach Aktivieren des Programms ihre bisherige Funktion ändern.



Da 'Multicolor' auf dem CPC 6128 geschrieben wurde, das Programm aber auf allen 3 CPCs laufen soll, sind die entsprechenden Änderungen für CPC 464 und 664 in den DATA-Zeilen enthalten.

(Christian Guensel/jf)

STERNENGRAFIKEN

Scheinbar simple Ideen führen oft zu erstaunlichen Ergebnissen. Das vorliegende Programm 'Sternengrafiken' gehört beispielsweise zu den Softwareprodukten, die durch ihre Variationsmöglichkeiten immer wieder zu beeindrucken vermögen.

Das Programm fragt bis zu drei Punkte mit x- und y-Koordinaten ab. Diese markierten Stellen sind dann die Ausgangspunkte für farbige Strahlen, die gerade durch ihr gegenseitiges Zusammenspiel fantastische, teilweise dreidimensional wirkende Strichlandschaften auf dem Bildschirm ergeben.

Selbstverständlich ist man in der Lage, die entstandenen Bilder auf Datenträger zu sichern.

Als Anregung nehmen Sie doch einmal folgende Werte:

Punkte: 3

Koordinaten: 320/40 320/60 320/80

Punkte 3

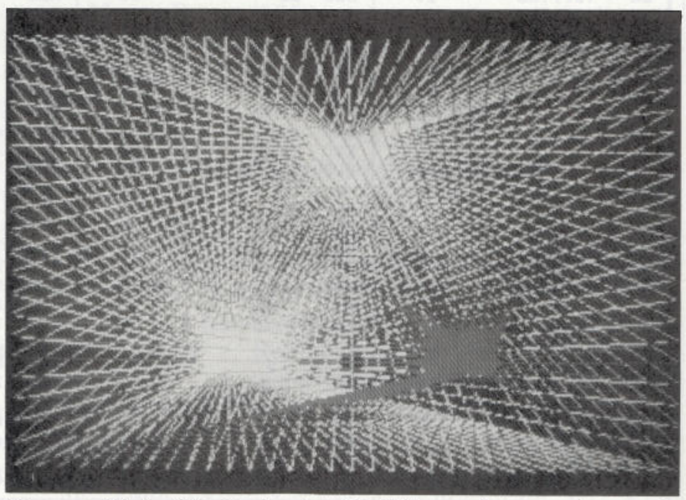
Koordinaten: 320/300 200/100 440/100

(Jörg Engel/jf)

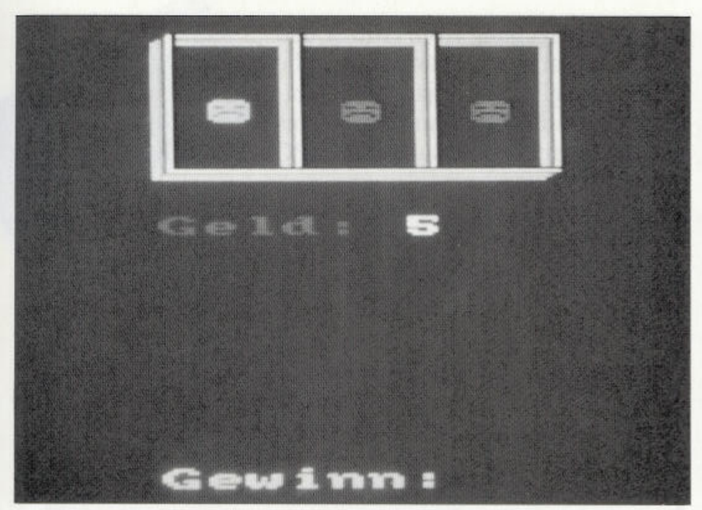
INKER

Dieses Programm soll helfen, die richtigen Inks für einen MODE 1-SCREEN zu finden. Es wurde eigentlich entwickelt, um gescannten Bildern eine vernünftige INK-Verteilung zuzuordnen, da beim Scannen häufig völlig unrealistische Farbzuordnungen auftreten. Es kann natürlich auch beim Einfärben eigener oder übernommener Screens hilfreich sein.

Der größte Nutzen tritt hierbei bei Monochrom-Monitoren auf, da die Screens aus professionellen Programmen in der



Sternengrafik besticht durch seine Flexibilität und erzeugt fantastische Bilder



Warum das Geld in die Casinos bringen? Die Herren und Damen sind reich genug. Simulieren Sie Ihren eigenen einarmigen Banditen

Regel für Farbmonitore ausgelegt sind und deshalb auf Grünmonitoren weniger ansprechend wirken.

Die Anwendung: Nach dem Starten gibt das Programm auf Tastendruck den Katalog der im Laufwerk liegenden Diskette aus, so daß man den Namen des Screens feststellen und eingeben kann. Nachdem dieser geladen ist, kann man mit dem Einfärben beginnen. Dazu dienen folgende Tasten:

1 -> INK 0
2 -> INK 1
3 -> INK 2
4 -> INK 3
I -> Feststellen der Ink-Zuteilung
F -> Find-Modus: Man kann einen Pen ausfindig machen
Q -> Rückkehr aus dem Find-Modus
E -> Laden eines neuen Screens
Bei I und F ist zu beachten, daß die unterste Zeile des Bildschirms überschrieben wird!

Haben Sie nach getaner Arbeit, sprich nach der harmonischen Zusammensetzung der Farben eines Bildes, die einzelnen Farbwerte notiert, so können Sie mit einem kleinen BASIC-Programm, das die Farbwerte setzt und anschließend das Bild einlädt, ihr erstelltes Werk jederzeit auf den Bildschirm holen.

(Karl Fischer/jf)

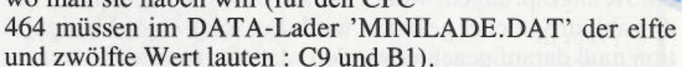
MINI-SCREEN

Daß sich in unserer Sparte "Kleinstprogramme" ein Werk wiederfindet, das nicht nur besonders klein ist, sondern auch noch in der Lage ist, Bilder zu verkleinern, trifft den Tenor dieser Rubrik in gewisser Weise doppelt – eine Tatsache, die natürlich besonders erfreulich ist.

Das Program "MINI-SCREEN" gibt ein Screen in einer Verkleinerung von einem Viertel des Originalscreens wieder.

Der Maschinencode wird ab &AF00 abgelegt, das Programm wird mit 'CALL &AF00,<adr>' gestartet, wobei die Variable <adr> die Adresse ist, ab welcher die verkleinerte Ausgabe erfolgt (ab &C000). Sollte zufällig oder absichtlich die Adresse kleiner als &C000 gewählt werden, dann sorgt das Programm automatisch dafür, daß die Verkleinerung in den Bildschirmspeicher kommt und nicht zum Beispiel den JUMP-Block des Betriebssystems überschreibt.

Im Mode 2 erkennt man am meisten vom ursprünglichen Bild, egal für welchen Modus das Bild erstellt worden ist. Das Programm berücksichtigt auch eventuelles Scrolling, so daß die Verkleinerung dahin kommt, wo man sie haben will (für den CPC 464 müssen im DATA-Lader 'MINILADE.DAT' der elfte und zwölfte Wert lauten : C9 und B1).



Das Orginalbild sollte ab &4000 abgespeichert sein, weil das Programm sich die Werte ab dieser Adresse holt. Wenn das Programm im Speicher ist, kann man mit 'POKE &AF80,&3F' die Verkleinerung invers ausgeben, ansonsten sollte &AF80 den Wert Null haben (der Wert &FF ergibt eine verlangsamte Ausgabe).

(Radoslaw Smiljanic/jf)

FAST-CLEAR

Bei FastClear handelt es sich um eine ungemein schnelle Routine, die Bildschirmausschnitte mit einem bestimmten Byte füllt. Solch eine Routine benötigt man zum Beispiel, um den Hintergrund eines Pulldown-Windows zu löschen.

Das Assemblerlisting umfaßt drei Routinen, die alle die gleichen Einsprungbedingungen haben und auch gleichartig funktionieren. Sie unterscheiden sich hauptsächlich durch die Länge und durch die Ausführungszeit.

Die erste Routine (TC) ist sichtbar 'langsam' (ähnlich dem CLG-Befehl im BASIC), die zweite (FCII) ist schon erheblich schneller, da sie nach dem neuen Prinzip funktioniert. Die dritte und schnellste Routine (FC) arbeitet nach dem Prinzip der zweiten, nur wird hier hauptsächlich mit selbstveränderlichem Code gearbeitet. Sie füllt den Bildschirm innerhalb von zirka 0.022 Sekunden.

Wieso ist die Routine so schnell?
Betrachtet man einmal die Z80-Befehle, die ein Byte in den Speicher schreiben, so ergeben sich folgende Möglichkeiten:

-LDIR 21 Taktzyklen für ein Byte
-LDI 16 Taktzyklen für ein Byte
-LD (HL),A 7 + 6 = 13 Taktzyklen für ein Byte
INC HL

Das sind die 'normalen' Befehle, wobei natürlich die Befehle LDI und LD (HL),A... noch eine Schleife 'drumherum' benötigen. Deshalb ist die Anzahl der Taktzyklen nicht ganz repräsentativ. Ein Befehl wurde in der Auflistung vergessen. Und genau dieser Befehl sorgt für die enorme Geschwindigkeit: **-PUSH DE 11 Taktzyklen für 2(!)Bytes**

Der PUSH-Befehl ist also mehr als doppelt so schnell wie die herkömmlichen Befehle. Um den PUSH-Befehl anwenden zu können, wurde folgendermaßen vorgegangen:

- Stapelzeiger zwischenspeichern
- Interrupt sperren
- Stapelzeiger hinter das Ende des gewünschten Adreßbereiches legen, da durch ein PUSH die Daten an- (SP-1) und (SP-2) abgelegt werden.
- Schleife mit PUSH-Befehl(en) ausführen





– alten Stapelzeiger zurückholen
– Interrupt wieder zulassen

Die Routine 'FC' ist hauptsächlich deshalb so schnell, weil die PUSH-Befehle nicht innerhalb einer Schleife ausgeführt werden, sondern 40 Befehle direkt hintereinander liegen. Die korrekte Anzahl wird dann über ein JR angesprungen, wobei der JR-Befehl entsprechend der Anzahl im Speicher geändert wird. Bei solch einer Manipulation muß darauf geachtet werden, daß die Anzahl der auszuführenden PUSHs im umgekehrten Verhältnis zum Sprungbefehl steht, das heißt, bei 40 PUSHs, die ausgeführt werden sollen, heißt der Befehl JR 0, bei 0 PUSHs heißt er JR 40. Daraus ergibt sich die Rechnung '40 minus Anzahl der P.'. Dieser Wert wird dann in den Sprungbefehl eingetragen.

Tippen Sie den Datalader ab, und starten Sie das Programm. Es erzeugt die Binärdatei 'FASTCLR.BIN". Mit

MEMORY &A4FF:LOAD "Fastclr.bin",&A500

bringen Sie das Programm in den Speicher. Der Aufruf der Routine erfolgt über

CALL &A500,x1,y1,x2,y2,byte

Der Wertebereich für die x-Variablen liegt zwischen 0 und 79, der für die y-Variablen zwischen 0 und 24. Der Koordinatenursprung ist in der oberen linken Ecke zu finden. Die Variable byte kann Werte zwischen 0 und 255 annehmen, die dann entsprechende Bytemuster in die Bildschirmausschnitte bringt.

(Andreas Stroiczek/jf)

CASINO

Nach soviel anstrengender Lektüre wollen wir Sie zur Unterhaltung in die glitzernden Hallen von Las Vegas führen. Da sich die Umgebung trotz Glitzer und Glimmer immer wieder als Quelle des finanziellen Verlustes entpuppt, sparen wir das Geld und geben uns der preiswerten Spielsucht am CPC hin.

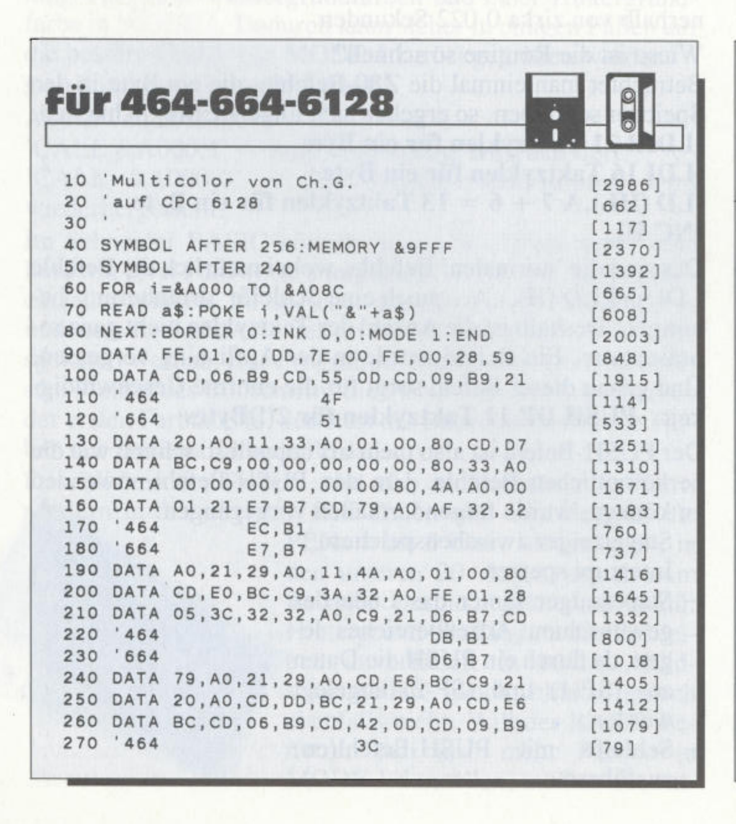
Das Programm Casino bringt Ihnen einen Spielautomaten auf den Monitor, mit dem Sie ebenso viel Freude haben können, wie bei einem Besuch in der Spielhalle. Ein fast typischer einarmiger Bandit wird simuliert. Das Ihnen zur Verfügung gestellte Startkapital kann bei entsprechenden Kombinationen erheblich gesteigert werden, und wenn es einmal nicht klappt, und das Geld den 'Bach runter geht', na, dann geben wir RUN ein, und das Vergnügen kann weitergehen.

(Steffen Moritz/jf)

Die Bedingungen

Hier die Bedingungen unseres Wettbewerbs: Wir suchen komplette Programme (bitte ohne Steuerzeichen!), die funktionsfähig möglichst auf allen drei CPCs laufen (was nicht heißen soll, daß wir bei besonders guten keine Ausnahme machen), und die nicht länger als 1 kByte sind (Anzeige auf dem Datenträger gilt!). Der Preis für solche Programme beträgt 100,– DM. Senden Sie uns Ihre Werke (maximal zwei Programme!) auf Datenträger (Diskette/Kassette) mit Programmbeschreibung (möglichst in ASCII-Form mit auf dem Datenträger) zu und zwar an den

DMV-Verlag
Postfach 250
Stichwort 100,– DM
3440 Eschwege



für 464-664-6128

```
10 'Multicolor von Ch.G.                 [2986]
20 'auf CPC 6128                         [662]
30 '                                     [117]
40 SYMBOL AFTER 256:MEMORY &9FFF         [2370]
50 SYMBOL AFTER 240                      [1392]
60 FOR i=&A000 TO &A08C                  [665]
70 READ a$:POKE i,VAL("&"+a$)            [608]
80 NEXT:BORDER 0:INK 0,0:MODE 1:END      [2003]
90 DATA FE,01,C0,DD,7E,00,FE,00,28,59    [848]
100 DATA CD,06,B9,CD,55,0D,CD,09,B9,21   [1915]
110 '464            4F                   [114]
120 '664            51                   [533]
130 DATA 20,A0,11,33,A0,01,00,80,CD,D7   [1251]
140 DATA BC,C9,00,00,00,00,80,33,A0      [1310]
150 DATA 00,00,00,00,00,00,80,4A,A0,00   [1671]
160 DATA 01,21,E7,B7,CD,79,A0,AF,32,32   [1183]
170 '464        EC,B1                    [760]
180 '664        E7,B7                    [737]
190 DATA A0,21,29,A0,11,4A,A0,01,00,80   [1416]
200 DATA CD,E0,BC,C9,3A,32,A0,FE,01,28   [1645]
210 DATA 05,3C,32,32,A0,C9,21,D6,B7,CD   [1332]
220 '464                        DB,B1    [800]
230 '664                        D6,B7    [1018]
240 DATA 79,A0,21,29,A0,CD,E6,BC,C9,21   [1405]
250 DATA 20,A0,CD,DD,BC,21,29,A0,CD,E6   [1412]
260 DATA BC,CD,06,B9,CD,42,0D,CD,09,B9   [1079]
270 '464                  3C             [79]
```

```
280 '664                  3E             [549]
290 DATA C9,01,01,7F,7E,E6,1F,F6,40,ED   [1903]
300 DATA 49,ED,79,23,0C,79,FE,04,20,F0   [1904]
310 DATA C9                              [276]
```

Listing MULTICOLOR

```
10 MODE 1:INK 0,0:INK 1,26:INK 2,15:INK 3, [3292]
6:PAPER 0:BORDER 0:PEN 1
20 LOCATE 14,3:PRINT"Sterngrafik":LOCATE 1 [4332]
4,4:PRINT"==========="
30 LOCATE 2,6:INPUT"Wieviele Punkte(max. 3 [4075]
):  ",anz
40 IF anz<1 OR anz>3 THEN 30               [1702]
50 FOR an=1 TO anz                         [428]
60 LOCATE 1,5+an*3:PRINT"X-Position Punkt" [5257]
;an;:INPUT"(10-630):  ",x(an)
70 IF x(an)<10 OR x(an)>630 THEN 60        [1659]
80 LOCATE 1,6+an*3:PRINT"y-Position Punkt" [3639]
;an;:INPUT"(10-390):  ",y(an)
90 IF y(an)<10 OR y(an)>390 THEN 80        [2338]
100 NEXT an                                [432]
105 LOCATE 1,18:PRINT"Absaven   ==> S":FOR [3018]
 t=1 TO 3000:NEXT t
110 CLS:xp=0:yp=0                          [1023]
120 FOR an=1 TO anz:pn=an                  [749]
130 FOR xp=1 TO 640 STEP 20:GOSUB 180:NEXT [2521]
 xp
140 FOR yp=1 TO 400 STEP 20:GOSUB 180:NEXT [2856]
 yp
150 FOR xp=640 TO 1 STEP -20:GOSUB 180:NEX [2904]
T xp
```

```
160 FOR yp=400 TO 1 STEP -20:GOSUB 180:NEX [2207]
T yp
170 NEXT an:GOTO 200                      [635]
180 MOVE x(an),y(an)                      [614]
190 DRAW xp,yp,pn:RETURN                  [1572]
200 WHILE INKEY$="":GOTO 200:WEND         [2051]
210 IF INKEY(60)=-1 THEN END              [1054]
220 SAVE "bild",b,&C000,&4000             [779]
```

Listing STERNENGRAFIKEN

```
10 ON ERROR GOTO 220                       [1529]
20 ON BREAK GOSUB 220                      [675]
30 e=0:f=8:g=16:h=24:INK 0,e:INK 1,f:INK 2 [3120]
,g:INK 3,h
40 MODE 2:CAT:PRINT:PRINT:CALL &BB03:INPUT [4929]
" Bildname:";n$
50 MODE 1:LOAD n$                          [1497]
60 SPEED KEY 1,1                           [1279]
70 IF INKEY$="1" THEN e=e+1:IF e=27 THEN e [1543]
=0
80 INK 0,e                                 [308]
90 IF INKEY$="2" THEN f=f+1:IF f=27 THEN f [1440]
=0
100 INK 1,f                                [307]
110 IF INKEY$="3" THEN g=g+1:IF g=27 THEN  [2737]
g=0
120 INK 2,g                                [282]
130 IF INKEY$="4" THEN h=h+1:IF h=27 THEN  [657]
h=0
140 INK 3,h                                [273]
150 IF INKEY$="i" THEN GOSUB 190           [815]
160 IF INKEY$="f"THEN GOSUB 240            [415]
170 IF INKEY$<>"e"THEN 70                  [641]
180 SPEED KEY 20,3:RUN                     [1511]
190 'Inkausgabe                            [489]
200 LOCATE 1,25:PRINT"Ink 0:";e;"Ink 1:";f [4055]
;"Ink 2:";g;"Ink 3:";h
210 RETURN                                 [555]
220 SPEED KEY 20,3                         [1378]
230 END                                    [110]
240 LOCATE 35,25:PRINT"F:"                 [1322]
250  a=VAL(INKEY$)-1:IF a<0 OR a>3 THEN 25 [3103]
0
260 LOCATE 37,25:PRINT a;                  [1300]
270 INK a,1,26:IF INKEY$<>"q" THEN 270     [2315]
280 INK 0,e:INK 1,f:INK 2,g:INK 3,h:LOCATE [4879]
 35,25:PRINT"   ";:RETURN
```

Listing INKER

```
1 DATA FE,1,C0,DD,5E,0,DD,56,1,2A,C4,B7,19 [12217]
,DD,21,0,40,6,19,18,5,11,50,0,DD,19,C5,DD,
E5,6,4,18,5,11,0,10,DD,19,C5,DD,E5,6,28,18
,5,23,DD,23,DD,23,C5,7C,FE,C0,30,6,11,50,C
0,19,18,F5,DD,4E,0,CD,7C,AF,DD,4E,1,CD,7C,
AF,C1,10,E0,DD,E1,11,20BA1
2 DATA D9,7,19,C1,10,CB,DD,E1,C1,10,BA,C9, [9711]
20,A4,20,62,79,20,52,41,44,4F,53,4C,41,56,
20,53,4D,49,4C,4A,41,4E,49,43,20,2C,20,31,
39,38,39,20,6,4,CB,21,0,CB,16,CB,21,10,F7,
C9,13962
3 FOR a=44800 TO 44936                     [841]
4 READ r$:IF LEN(r$)>2 THEN 6              [1816]
5 POKE a,VAL("&"+r$):s1=s1+PEEK(a):s2=s2+P [3072]
EEK(a):NEXT
6 v=VAL("&"+LEFT$(r$,4)):z=VAL("&"+RIGHT$( [12737]
r$,LEN(r$)-4)):PRINT CHR$(12)z:IF v<>s1 TH
EN PRINT"DATA-Zeile"z"falsch!":END ELSE IF
 a<44936 THEN s1=0:GOTO 4 ELSE IF s2<>1339
2 THEN PRINT"Totale Summe falsch!"
7 SAVE"mini.bin",b,&AF00,&88,&AF00         [1795]
```

Listing MINI-SCREEN

```
10 '====================================== [2875]
==============
20 '= Dieser Datalader erzeugt die Datei ' [3373]
FASTCLR.BIN' =
30 '====================================== [2875]
==============
40 MODE 2                                  [513]
50 DEFINT a-z                              [553]
60 RESTORE 200                             [563]
70 b=8:length=&D4                          [885]
80 z=length\b-((length MOD b)>0)           [2914]
90 zeile=200                               [653]
100 FOR i=&A500 TO &A500+(z-1)*b STEP b    [1232]
110 FOR j=i TO i+b-1                       [857]
120 READ a$:a=VAL("&"+a$)                  [883]
130 POKE j,a                               [153]
140 pr=(pr+(j-i+1)*a)MOD 4096              [977]
150 NEXT                                   [350]
160 READ a$:pr2=VAL("&"+a$)                [414]
170 IF pr2<>pr THEN PRINT"Pruefsummenfehle [2759]
r in "zeile:STOP
180 zeile=zeile+1:NEXT                     [1002]
190 SAVE"FastClr.BIN",b,&A500,length:END   [2441]
200 DATA FE,05,C0,4B,26,00,54,DD,E66       [1466]
201 DATA 7E,06,FE,19,38,02,3E,18,5E4       [2206]
202 DATA 6F,DD,7E,02,FE,19,38,02,0B3       [2110]
203 DATA 3E,18,5F,CD,50,A5,D5,7D,49B       [1651]
204 DATA 93,47,04,CB,20,CB,20,CB,58E       [1059]
205 DATA 20,DD,7E,08,FE,50,38,02,170       [1539]
206 DATA 3E,4F,6F,DD,7E,04,FE,50,30D       [1971]
207 DATA 38,02,3E,4F,5F,CD,50,A5,340       [1400]
208 DATA 7D,E1,63,93,59,4F,0C,7B,0B7       [2108]
209 DATA F5,CD,57,A5,F1,C3,6C,A5,842       [1931]
210 DATA E5,ED,52,E1,D0,EB,C9,16,F3C       [1164]
211 DATA 00,7D,87,87,87,17,CB,12,D31       [1769]
212 DATA 5F,7C,62,6B,29,29,19,5F,6C4       [1260]
213 DATA 16,C0,19,C9,5F,78,B7,C8,BB5       [1778]
214 DATA 79,B7,C8,F3,ED,73,D2,A5,5F9       [1296]
215 DATA 50,06,00,09,79,1F,30,03,AF8       [1063]
216 DATA 2B,06,73,D6,28,ED,44,32,9A2       [1561]
217 DATA 93,A5,78,42,53,32,90,A5,9D2       [1152]
218 DATA 73,F9,18,FE,D5,D5,D5,D5,619       [785]
219 DATA D5,D5,D5,D5,D5,D5,D5,D5,40D       [1336]
220 DATA D5,D5,D5,D5,D5,D5,D5,D5,201       [1399]
221 DATA D5,D5,D5,D5,D5,D5,D5,D5,FF5       [1671]
222 DATA D5,D5,D5,D5,D5,D5,D5,D5,DE9       [1689]
223 DATA D5,D5,D5,D5,7C,C6,08,67,0BB       [960]
224 DATA 30,08,7D,C6,50,6F,7C,CE,388       [2163]
225 DATA C0,67,10,C4,ED,7B,D2,A5,ABF       [686]
226 DATA FB,C9,00,00,00,00,00,00,D4C       [1440]
```

Listing FAST-CLEAR

```
10 MODE 0: FOR i=0 TO 6:READ a:INK i,a:NEX [11191]
T:FOR k=4 TO 20 STEP 4:MOVE 152+k,250+k/2:
DRAWR 310,0,k/4:DRAWR 0,100:DRAWR-310,0:DR
AWR 0,-100:MOVER 100,0:DRAWR 0,100:MOVER 1
10,-100:DRAWR 0,100:NEXT:m=25:PEN 6:LOCATE
 6,12
11 DATA 0,25,24,15,6,3,18                  [885]
20 PRINT"Geld:"m:PEN 2:LOCATE 6,24:PRINT"G [2356]
ewinn:":GOSUB 100
30 TAG:x=0:y=0                             [741]
40 IF x=0 THEN PLOT 200,310,RND*3+1:PRINT  [3729]
CHR$(223+a);:a=a MOD 8+1
50 IF y=0 THEN PLOT 305,310,RND*3+1:PRINT  [3014]
CHR$(223+b);:b=b MOD 8+1
60 PLOT 410,310,RND*3+1:PRINT CHR$(223+c); [3037]
:c=c MOD 8+1
70 IF INKEY$<>""THEN IF x=0 THEN x=1 ELSE  [3401]
IF x=1 THEN y=1:x=2 ELSE IF y=1 THEN GOTO
90
80 SOUND 1,9,1,,,,19:GOTO 40               [2120]
90 IF a=b AND a=c THEN w=n*a*10:m=m+w:GOSU [11347]
B 130:GOSUB 110:GOSUB 120:GOSUB 100:GOTO 3
0 ELSE IF a=b OR a=c OR b=c THEN w=2*n:m=m
+w:GOSUB 130:GOSUB 110:GOSUB 120:GOSUB 100
:GOTO 30 ELSE w=0:GOSUB 130:GOSUB 100:GOTO
 30
100 LOCATE 6,18:IF m<1 THEN PEN 1:PRINT"Ge [12995]
ld alle!":CALL &BB18:RUN ELSE PEN 4:INPUT"
Einsatz";n:IF m-n<0 THEN 100 ELSE IF n=0 T
HEN PEN 1:LOCATE 5,18:PRINT"Endgewinn:";m-
25:CALL &BB18:RUN ELSE LOCATE 6,18:PRINT S
PACE$(14):a=1:b=1:c=1:m=m-n:GOSUB 110:RETU
RN
110 LOCATE 11,12:PEN 4:PRINT m"     ":FOR  [5867]
k=15 TO 8 STEP-1:SOUND 1,160,5,k:NEXT:RETU
RN
120 FOR k=800 TO 0 STEP-2:SOUND 129,k,1:NE [1741]
XT k:RETURN
130 TAGOFF:PEN 3:LOCATE 13,24:PRINT w"     [2776]
 ":RETURN
```

Listing CASINO

Kleine Ursache – große Wirkung



Mit wenigen POKEs, CALLs oder OUTs BASIC-Programme 'stylen'

Das ROM und RAM des CPC ist eine wahre Fundgrube für den Maschinenspracheprogrammierer. Viele oft benutzte Assemblerroutinen sind schon vorhanden und müssen vom Programmierer nicht noch einmal programmiert werden. Einige Routinen und Speicherstellen sind auch von BASIC aus ansprechbar und können oft Programme ersetzen, die in BASIC entweder zu lang wären oder überhaupt nicht programmiert werden könnten. Dieser Artikel stellt eine ganze Reihe CALLs, POKEs, OUTs vor, die sicher nützlich sind oder auch nur als kleiner Grafikgag dienen sollen.

Die angegebenen Adressen sind immer hexadezimal angegeben und beziehen sich grundsätzlich auf alle CPCs. Eventuelle Unterschiede beim CPC 464 werden angegeben.

Grafik

POKE &B6A3,x (CPC 464: &B338)
Mit diesem POKE wird das Grafikmuster angegeben, das beim Linienzeichnen verwendet werden soll. Es ist dem MASC-Befehl bei den CPCs 664/6128 sehr ähnlich.

POKE &B72F,x (CPC 464: &B28F)
Dieser POKE stellt das Muster ein, mit dem das Bildschirmzeichen gezeichnet werden soll.

POKE &B730,x (CPC 464: &B290)
Dieser POKE stellt das Muster ein, mit dem der Hintergrund des Bildschirmzeichens gezeichnet werden soll.

CALL &BB81
Diese Routine gibt bei INPUT-, INKEY- und ähnlichen Befehlen den Cursor frei.

CALL &BB84
Diese Routine sperrt den durch CALL &BB81 freigegebenen Cursor wieder.

CALL &BB9C
Durch diesen Befehl wird PEN und PAPER analog zum ASCII-Zeichen 24 getauscht.

CALL &BC00
Dieses Kommando setzt alle Farben auf den Normalzustand.

CALL &BC06,x (x = &40 oder &C0)
Mit diesem POKE wird der Offset des Bildschirms gesetzt. Aufgrund der Speicheraufteilung des CPC sind aber nur zwei Bildschirmoffsets möglich: der normale ab &C000 und der zweite ab &4000. Mit &C0 oder &40 als x-Wert werden diese entspechend eingeschaltet. Andere Offsets führen mit großer Sicherheit zum Absturz des Systems.

CALL &BC40
Dieses Kommando scrollt den Bildschirm hardwaremäßig nach unten.

POKE &BDD0,&C9
Zwischendurch mal ein Grafikgag! Geben Sie doch bitte diesen POKE ein und drücken Sie einige Male auf die Cursor-Tasten. Sie werden sehen, was passiert!

POKE &BDEB,&C9
Nach diesem POKE wird der Bildschirm nach einem MODE-Befehl nicht mehr gelöscht. Interessante Möglichkeiten tun sich auf…

dazu noch einige OUTs…

OUT &BC00,1:OUT &BD00,Spalten
Diese zwei OUTS geben an, wie viele Spalten auf dem Bildschirm dargestellt werden. Der Standardwert beträgt 40.

OUT &BC00,2:OUT &BD00,x
Mit diesem Befehl kann ein horizontaler Scroll erreicht werden. Standardwert für x ist 46. Abweichungen von dieser Zahl scrollen den Bildschirm horizontal in die entsprechende Richtung. Zu beachten ist allerdings, daß sich die Bildschirmadressen auch vertikal verschieben können!

OUT &BC00,6:OUT &BD00,Zeilen
Diese OUTs geben an, wie viele Zeilen auf dem Monitor dargestellt werden. Zu beachten ist allerdings, daß diese Zeilen intern weiterexistieren. Standardwert ist 25.

OUT &BC00,7:OUT &BD00,x
Mit diesem Befehl kann ein vertikaler Scroll erreicht werden. Standardwert für x ist 30. Abweichungen von dieser Zahl scrollen den Bildschirm vertikal in die entsprechende Richtung. Zu beachten ist allerdings, daß sich die Bildschirmadressen auch horizontal verschieben können!

Diskettenlaufwerk

POKE &A701,User-Nr. (Vortex F1-X: &A705)
Mit diesem Befehl wählen Sie die User-Nummer des Diskettenlaufwerks.

POKE &A701,&E5:CAT
Dieses Kommando kann in Notfällen ganz schön nützlich sein! Es zeigt die gelöschten Dateien auf der Diskette an. Diese können Sie dann laden und bei User 0 abspeichern. Voraussetzung für diese File-Rettung ist allerdings, daß zwischenzeitlich nichts auf der Diskette abgespeichert wurde!

OUT &FA7E,1
Dieser OUT-Befehl schaltet das Diskettenlaufwerk ein.

OUT &FA7E,0
Dieses OUT-Kommando schaltet das Diskettenlaufwerk wieder aus.

Diverse POKEs und CALLs...

CALL 0
Durch diesen CALL wird ein sogenannter Reset ausgelöst.

POKE 370,0:POKE &371,0
Diese beiden POKEs machen die erste Zeile in einem BASIC-Programm unsichtbar. Sprünge auf diese Zeile sind allerdings auch nicht mehr möglich. Mit **POKE 370,1** erhält die unsichtbare erste Zeile wieder die Nummer 1. Sie ist dann wieder sichtbar.

POKE &AC00,1
Wenn Sie nach diesem POKE ein BASIC-Programm eintippen, werden überflüssige Leerzeichen entfernt. Das hilft, Speicherplatz zu sparen.

POKE &AC00,0
schaltet den Komprimiermodus wieder ab.

POKE &B632,1 (CPC 464:&B4E8)
Dieser Befehl schaltet auf Großschreibweise.

POKE &B632,0 (CPC 464: &B4E8)
Dieses Kommando schaltet auf Kleinschreibweise.

CALL &BB06 oder **CALL &BB18** Nach diesem CALL wartet der CPC auf einen Tastendruck.

POKE &BB5A,&C9
Dieser POKE schaltet die Bildschirmausgabe aus. Durch **POKE &BB5A,&CF** kann sie wieder eingeschaltet werden.

POKE &BB5B,&1B:POKE &BB5C,&88.
Diese beiden POKEs leiten alle Bildschirmausgaben zum Drucker.

(Markus Felder/cd)

Ein Drucker lernt schreiben

Schreibschrift auf dem LQ3500

Drucker bieten viele Möglichkeiten, mit verschiedenen Schriften in verschiedenen Größen zu schreiben. Auch der LQ bietet dies und noch etwas mehr an.

Er erlaubt es dem Anwender, selbst Zeichen, ja ganze Zeichensätze zu entwerfen, die dann mehr oder weniger gute Ergebnisse auf dem Drucker erzielen. Einen wirklich gelungenen Zeichensatz, der es Ihnen ermöglicht, Ihre Texte in 'Schreibschrift' auszugeben, möchten wir Ihnen hier vorstellen.

Abtippen

Tippen Sie zuerst Listing 1 ein und speichern es dann unter dem Namen SCRIPT.BAS ab. Haben Sie das erledigt, können Sie noch einmal einen Test mit dem Checksummer durchführen und sich nun an das zweite Listing begeben. Hier sollten Sie besonders genau aufpassen, daß die Checksummen stimmen, da das Programm keine Sicherungsroutine beinhaltet. Speichern Sie dieses Programm nun unter dem Namen SCRIPT.FON auf Ihrer Diskette oder Kassette ab.

Nach dem Start des Programms mit RUN"SCRIPT [ENTER] wird nach einer Sicherheitsabfrage das Programm SCRIPT.FON nachgeladen, und die Umdefinition der einzelnen Zeichen beginnt. Meldet sich der Computer mit READY, können Sie ein beliebiges Programm starten, und der Ausdruck kann (in Schreibschrift) beginnen.

Zeichenkunde

Wenn Sie selbst Zeichen definieren möchten, kann Ihnen dies sicherlich von Nutzen sein.

Die Zeichenmatrix des LQ sieht gegenüber der des normalen CPC etwas unterschiedlich aus. Sie unterscheidet sich schon allein in der Rastergröße für die einzelnen Zeichen. Gegenüber dem CPC (8x8) hat der LQ eine Auflösung von 24x11. Jeder Buchstabe besteht also aus elf Reihen von Einzelpunkten, wobei jede Reihe in drei Gruppen mit acht Punkten (3*8 = 24 Nadeln) unterteilt ist.

Der Buchstabe ist von links nach rechts und von oben nach unten durch 33 Zahlen zu definieren.

Die DATA-Zeile (in unserem Programm) sieht also wie folgt aus:

DATA A1,B1,C1,A2,B2,C2,A3 ... A11,B11,C11,X

Der 34. Wert 'X' in dieser Zeile veranlaßt das Programm, auf die Definition des nächsten Zeichens überzuwechseln.

Die Werte A1 bis C11 werden durch Addition der Punktwerte (8 Punkte pro Gruppe) errechnet.

Und jetzt ran an die Arbeit: Entwerfen Sie eigene Schriften!

(Stefan Wegstein/rs)

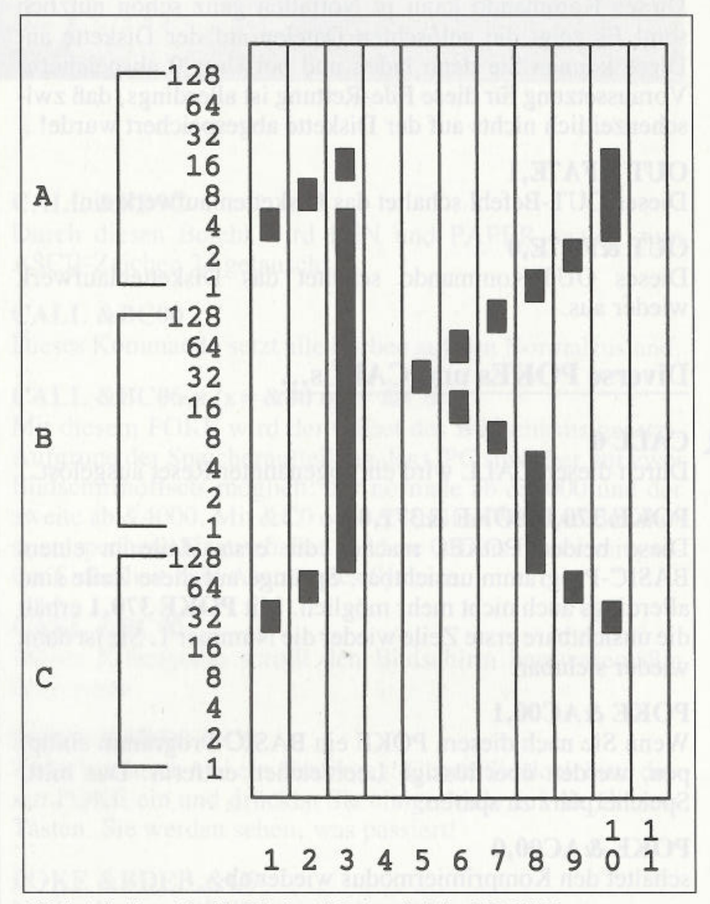


Abb1:Aufteilung für die Zeichendefiniton bei dem LQ 3500

für 464-664-6128

```
1 ' Ladeprogramm fuer Download-Zeichensatz [2383]
2 ' ---------------------------------------- [1873]
3 '  1989 By Stefan Wegstein                [377]
4 '                                         [117]
5 '                                         [117]
10 MODE 2:CLS                               [1257]
20 ORIGIN 0,0:DRAWR 639,0:DRAWR 0,399:DRAW [2696]
 0,399:DRAW 0,0
```

Listing DOWNLOAD

```
30 LOCATE 10,3:FOR d=1 TO 19: READ n:PRINT [3196]
 CHR$(n);:NEXT
40 DATA 150,154,154,156,32,32,32,32,32,32, [3321]
32,32,32,32,32,32,32,156,32
50 LOCATE 10,4:FOR d=1 TO 19: READ n:PRINT [3652]
 CHR$(n);:NEXT
60 DATA 149,32,32,32,32,32,32,32,32,32,32, [2080]
144,32,32,32,32,32,149,32
70 LOCATE 10,5:FOR d=1 TO 19: READ n:PRINT [2906]
 CHR$(n);:NEXT
80 DATA 147,154,154,156,150,154,156,32,150 [3550]
,154,153,148,32,150,154,156,146,159,152
```

Listing DOWNLOAD

```
90 LOCATE 10,6:FOR d=1 TO 19: READ n:PRINT [3014]
 CHR$(n);:NEXT
100 DATA 32,32,32,149,149,32,32,32,149,32, [3031]
32,149,32,149,32,149,32,149,32
110 LOCATE 10,7:FOR d=1 TO 19: READ n:PRIN [3294]
T CHR$(n);:NEXT
120 DATA 147,154,154,153,147,154,153,32,14 [3220]
5,32,32,147,32,151,154,153,32,147,32
130 LOCATE 10,8:FOR d=1 TO 19: READ n:PRIN [2311]
T CHR$(n);:NEXT
140 DATA 32,32,32,32,32,32,32,32,32,32,32, [2334]
32,32,153,32,32,32,32,32
150 LOCATE 35,6:PRINT "Download-Zeichensat [2951]
z fuer LQ 3500/"
160 LOCATE 35,7:PRINT "CPC 464, 664, 6128" [2430]
170 ORIGIN 0,250:DRAW 639,0                [1102]
180 LOCATE 10,12:PRINT "Bitte ueberpruefen [7682]
 Sie ob Ihr Drucker richtig eingestellt is
t.
190 LOCATE 10,14:PRINT "- DIP-Schalter 2-4 [3393]
 auf <ON>"
200 LOCATE 10,15:PRINT "- DIP-Schalter 1-2 [1663]
 auf <ON>"
210 LOCATE 10,16:PRINT "- Drucker auf      [2272]
<ON LINE>"
220 LOCATE 10,18:PRINT "Drucker bereit (j) [6134]
    entsprechende Taste druecken"
230 LOCATE 10,22:PRINT "Papier einlegen, A [3943]
ndruckbuegel schliessen"
240 a$=INKEY$: IF a$<>"j" THEN GOTO 180    [1206]
250 IF a$="j" THEN GOTO 260                [1523]
260 CLS                                    [91]
270 LOCATE 12,5:PRINT "B i t t e   w a r   [2205]
t e n"
280 LOCATE 12,7:PRINT "Zeichensatz wird je [4165]
tzt geladen."
290 LOCATE 12,8:PRINT "Wenn akustisches Si [6266]
gnal ertoent ist Drucker bereit"
300 RUN "script.fon"                       [700]
```

```
10 ' Download-Zeichensatz "Script"         [3486]
20 ' ---------------------------------     [1785]
30 ' 1989 By Stefan Wegstein               [377]
40 '                                       [117]
50 '                                       [117]
60 WIDTH 255                               [957]
70 x=32                                    [526]
80 PRINT #8,; CHR$(27)"&"CHR$(0)CHR$(x)CHR [3650]
$(x)CHR$(0)CHR$(11)CHR$(2);
90 x=x+1                                   [853]
100 FOR d=1 TO 34:READ n:PRINT #8,; CHR$(n [6564]
);:NEXT:IF n=22 GOTO 80:IF n=99 GOTO 1020
110 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 [3966]
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,22
120 DATA 0,0,0,0,0,0,0,0,0,0,15,252,16 [4299]
,16,2,32,15,252,16,0,0,0,0,0,0,0,0,0,
    0,0,22
130 DATA 0,0,0,0,0,0,0,0,31,0,0,0,0,0,0, [3765]
0,0,0,0,0,31,0,0,0,0,0,0,0,0,0,0,22
140 DATA 0,66,0,0,0,0,0,66,0,15,189,224,0, [4341]
66,0,0,0,0,0,66,0,15,189,224,0,66,0,0,
    0,0,0,66,0,22
150 DATA 0,0,0,1,194,0,2,33,0,4,16,128,29, [3904]
207,96,0,16,128,29,207,96,4,16,128,2,
    9,0,1,6,0,0,0,0,22
160 DATA 0,0,0,16,0,0,8,0,96,16,1,128,0,14 [4446]
,0,0,112,0,3,128,0,28,0,32,0,0,64,0,0,
    32,0,0,0,22
170 DATA 0,0,0,0,7,128,3,232,64,4,16,32,8, [5060]
40,0,16,68,32,0,130,64,17,1,0,14,0,
    128,0,2,64,0,4,32,22
180 DATA 0,0,0,0,0,0,0,0,0,0,0,0,64,0,13 [3198]
,128,0,18,0,0,12,0,0,0,0,0,0,0,0,0,
    0,22
190 DATA 0,0,0,0,124,0,0,130,0,1,1,0,2,0,1 [4626]
28,4,0,64,8,0,32,0,0,0,0,0,0,0,0,0,
    0,0,22
200 DATA 0,0,0,0,0,0,0,0,0,0,8,0,32,4, [3400]
0,64,2,0,128,1,1,0,0,130,0,0,124,0,0,
    0,0,22
210 DATA 0,16,0,1,1,0,0,146,0,0,68,0,0,56, [4959]
0,15,199,192,0,56,0,0,68,0,0,146,0,1,
    1,0,0,16,0,22
220 DATA 0,16,0,0,0,0,0,16,0,0,0,0,0,16,0, [4627]
3,239,128,0,16,0,0,0,0,0,16,0,0,0,0,
    0,16,0,22
230 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,1,144,0 [3709]
,2,32,0,1,192,0,0,0,0,0,0,0,0,0,0,
    0,22
240 DATA 0,0,0,0,16,0,0,0,0,0,16,0,0,0,0,0 [3215]
,16,0,0,0,0,0,16,0,0,0,0,0,16,0,0,0,
    0,22
250 DATA 0,0,0,0,0,0,0,0,0,0,16,0,0,32,0 [4294]
,0,16,0,0,0,0,0,0,0,0,0,0,0,0,0,0,22
260 DATA 0,0,0,0,0,96,0,0,128,0,3,0,0,4,0, [3940]
0,24,0,0,32,0,0,192,0,1,0,0,14,0,0,0,
    0,0,22
270 DATA 0,0,0,3,255,128,4,0,64,8,0,32,0,0 [4398]
,0,8,0,32,0,0,0,8,0,32,4,0,64,11,255,
    128,16,0,0,22
280 DATA 0,0,0,0,32,0,0,64,0,0,128,0,1,0,0 [3642]
,2,0,0,4,0,0,8,0,0,0,0,0,15,255,224,
    0,0,0,22
290 DATA 0,0,0,1,128,96,2,64,128,4,1,32,8, [4521]
2,64,0,4,0,8,8,64,0,16,32,8,32,0,15,
    192,32,0,0,0,22
300 DATA 0,0,0,1,0,128,2,0,64,4,16,32,8,0, [4043]
0,0,16,32,8,40,0,0,68,32,8,130,64,15,
    3,128,0,0,0,22
310 DATA 0,0,0,0,0,0,1,248,0,2,0,0,4,8,0,8 [3780]
,0,0,0,8,0,0,119,224,0,8,0,0,0,0,0,8,
    0,22
320 DATA 0,0,0,15,224,192,0,0,32,8,32,0,4, [5581]
0,32,0,32,0,4,0,32,0,32,64,4,16,128,
    8,15,0,0,0,0,22
330 DATA 0,0,0,1,255,128,2,0,64,4,7,32,8,8 [5336]
,0,0,0,32,8,8,0,4,0,32,2,8,64,0,7,128,
    0,0,0,22
340 DATA 0,0,0,8,0,0,4,8,0,0,0,32,4,8,192, [4576]
0,3,0,4,12,0,8,48,0,0,196,0,15,0,0,0,
    8,0,22
350 DATA 0,0,0,3,143,128,4,80,64,8,32,32,0 [5052]
,0,0,8,32,32,0,0,0,8,32,32,4,80,64,3,
    143,128,0,0,0,22
360 DATA 0,0,0,3,128,128,4,64,64,8,32,32,0 [5122]
,0,0,8,32,32,0,0,0,8,.32,32,4,0,64,3,
    255,128,0,0,0,22
370 DATA 0,0,0,0,0,0,2,0,32,1,0,64,2,0,32, [4540]
0,0,0,0,0,0,0,0,0,0,0,0,0,.0,0,0,0,22
380 DATA 0,0,0,0,0,0,2,1,16,1,2,32,2,1,192 [4103]
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    22
390 DATA 0,0,0,0,0,0,16,0,0,40,0,0,68,0, [4052]
0,130,0,1,1,0,2,0,128,4,0,64,8,0,32,0,
    0,0,22
400 DATA 0,0,0,0,66,0,0,0,0,0,66,0,0,0,0,0 [3780]
,66,0,0,0,0,0,66,0,0,0,0,0,66,0,0,0,0,
    22
410 DATA 0,0,0,8,0,32,4,0,64,2,0,128,1,1,0 [4196]
,0,130,0,0,68,0,0,40,0,0,16,0,0,0,0,0,
    0,0,22
420 DATA 0,0,0,3,0,0,4,0,0,8,0,0,0,0,16,8, [3638]
7,32,0,8,16,8,16,0,4,32,0,3,192,0,0,0,
    0,22
430 DATA 0,0,0,1,255,192,2,0,32,4,0,16,8,5 [4744]
6,0,0,68,16,8,0,0,4,68,16,2,8,0,1,240,
    16,0,0,0,22
440 DATA 0,0,128,0,0,64,0,12,32,0,18,192,0 [3855]
,9,0,1,244,0,14,3,128,16,0,0,15,255,
    192,0,0,32,0,0,64,22
450 DATA 16,0,64,0,0,32,31,255,192,0,0,0,1 [5289]
6,32,64,0,80,32,16,136,0,9,4,32,6,3,
    192,0,0,0,0,0,22
460 DATA 1,0,0,0,64,0,3,191,128,4,64,64,8, [3333]
```

Listing DOWNLOAD

Listing DOWNLOAD

```
0,32,16,64,0,0,130,32,17,1,64,10,0,
     128,4,0,0,0,0,0,22
470 DATA 4,0,128,10,0,64,16,0,32,0,0,31, [3967]
255,224,0,0,0,16,0,32,8,0,64,4,1,128,
     1,254,0,0,0,0,22
480 DATA 3,131,128,4,68,64,8,40,32,16,16,0 [4856]
,0,0,32,16,16,0,0,0,32,16,16,64,8,0,
     128,4,1,0,0,0,0,22
490 DATA 3,0,128,4,128,64,8,0,32,16,4,0,0, [4437]
8,32,16,0,64,15,255,128,16,0,0,0,8,0,
     16,16,0,0,0,0,22
500 DATA 2,0,0,0,64,0,3,191,132,4,64,66,8, [3786]
0,33,16,64,0,0,128,33,17,0,66,10,3,
     188,4,0,0,0,0,0,22
510 DATA 4,2,32,8,4,64,23,251,128,0,0,0,0, [3826]
8,0,0,0,0,0,8,0,7,247,192,8,32,32,16,
     64,64,0,0,128,22
520 DATA 0,0,0,6,0,0,9,0,128,16,0,64,0,0,3 [4702]
2,16,0,0,0,0,32,16,0,64,15,255,128,16,
     0,0,0,0,0,22
530 DATA 0,0,0,6,0,0,9,0,14,16,0,17,0,0,32 [4362]
,16,0,65,0,0,130,16,1,4,15,254,248,16,
     2,0,0,4,0,22
540 DATA 4,0,32,8,0,64,15,255,128,0,0,0,0, [4235]
32,0,0,80,0,0,136,0,1,7,128,2,0,64,28,
     0,32,0,0,0,22
550 DATA 1,0,192,0,129,32,0,64,0,0,1,32,0, [3965]
64,192,7,191,0,8,64,64,16,128,32,1,0,
     0,18,0,32,12,0,64,22
560 DATA 0,0,64,0,0,32,15,255,192,16,0,0,1 [4268]
2,0,0,3,0,0,0,248,0,3,0,0,12,0,0,16,0,
     0,15,255,224,22
570 DATA 0,0,64,0,0,32,15,255,192,16,0,0,1 [5155]
5,128,0,0,124,0,0,3,192,0,0,32,15,255,
     192,16,0,0,8,0,0,22
580 DATA 0,0,0,7,255,128,8,0,64,23,192,32, [4514]
8,64,0,16,32,32,0,0,0,16,0,32,8,0,64,
     7,255,192,0,0,0,22
590 DATA 0,0,0,16,64,64,0,16,32,31,239,192 [4136]
,0,16,0,16,0,0,0,16,0,16,32,0,8,64,0,
     7,128,0,0,0,0,22
600 DATA 0,0,0,7,255,128,8,0,64,16,0,32,0, [4527]
12,0,16,4,32,0,1,0,16,0,32,8,0,64,7,
     255,128,0,0,28,22
610 DATA 16,0,64,0,0,32,31,255,192,0,0,0,1 [4550]
6,32,0,0,88,0,16,134,0,9,1,192,6,0,32,
     0,0,64,0,0,128,22
620 DATA 1,0,0,0,128,128,0,64,64,0,0,32,0, [4773]
64,0,7,128,32,8,64,0,16,160,32,1,31,
     192,18,0,0,12,0,0,22
630 DATA 3,0,128,4,128,64,8,0,32,16,0,0,0, [4825]
0,32,16,0,64,15,255,128,16,0,0,0,0,0,
     16,0,0,0,0,0,22
640 DATA 8,0,0,16,0,0,15,255,128,0,0,64,0, [3360]
0,32,14,0,0,16,0,32,0,0,0,16,0,32,15,
     255,192,0,0,0,22
650 DATA 8,0,0,16,0,0,15,248,0,0,6,0,0,1,0 [3375]
,1,0,192,2,0,32,4,0,192,8,1,0,16,6,0,
     15,248,0,22
660 DATA 31,255,192,0,0,32,0,0,64,0,1,128, [3365]
0,61,0,0,1,128,0,0,64,0,0,32,15,255,
     192,16,0,0,8,0,0,22
670 DATA 6,1,128,8,0,64,16,0,32,8,0,64,4,0 [4240]
,128,3,255,0,4,0,128,8,0,64,16,0,32,8,
     0,64,6,1,128,22
680 DATA 16,0,64,8,0,32,4,0,0,2,0,32,1,0,6 [4188]
4,0,255,128,1,0,0,2,0,0,4,0,0,8,0,0,
     16,0,0,22
690 DATA 4,0,96,10,0,128,16,17,32,0,2,0,16 [5178]
,20,32,0,8,0,16,16,32,0,96,0,17,144,
     32,6,0,64,24,16,128,22
700 DATA 0,0,0,0,0,0,31,255,224,0,0,0,16,0 [3563]
,32,0,0,0,16,0,32,0,0,0,16,0,32,0,0,
     0,0,0,0,22
710 DATA 0,0,0,28,0,0,2,0,0,1,128,0,0,64,0 [3477]
,0,48,0,0,8,0,0,6,0,0,1,0,0,0,192,0,
     0,32,22
720 DATA 0,0,0,0,0,0,16,0,32,0,0,0,16,0,32 [4846]
,0,0,0,16,0,32,0,0,0,31,255,224,0,0,
     0,0,0,0,22
730 DATA 0,0,0,1,0,0,2,0,0,5,0,0,8,0,0,17, [3700]
255,224,8,0,0,5,0,0,2,0,0,1,0,0,0,0,
     0,22
740 DATA 0,0,8,0,0,0,0,0,8,0,0,0,0,0,8,0,0 [3044]
,0,0,0,8,0,0,0,0,0,8,0,0,0,0,0,8,22
```

Listing DOWNLOAD

```
750 DATA 0,0,0,0,0,0,16,0,0,8,0,0,4,0,0,2, [4006]
0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,22
760 DATA 0,15,128,0,16,64,0,32,32,0,0,0,0, [5222]
32,32,0,0,64,0,32,128,0,31,64,0,32,
     32,0,0,0,0,0,32,22
770 DATA 8,0,32,16,0,64,15,255,128,0,0,96, [4781]
0,16,0,0,32,32,0,0,0,0,32,32,0,16,64,
     0,15,128,0,0,32,22
780 DATA 0,0,32,0,0,64,0,15,128,0,16,64,0, [4318]
32,32,0,0,0,0,32,32,0,0,64,0,32,128,
     0,24,0,0,0,0,22
790 DATA 0,15,128,0,16,64,0,32,32,0,0,0,0, [4382]
32,32,0,0,0,0,32,32,15,223,192,16,0,
     32,8,0,0,0,0,32,22
800 DATA 0,0,32,0,0,0,0,0,32,0,14,64,0,17, [4117]
128,0,32,64,0,2,32,0,36,0,0,24,32,0,
     0,0,0,0,32,22
810 DATA 0,0,32,0,0,0,0,0,32,0,0,64,7,251, [4437]
191,8,2,64,16,4,32,8,8,0,7,240,32,0,
     0,0,0,0,0,22
820 DATA 0,0,64,0,0,32,0,15,128,0,16,66,0, [4704]
32,33,0,0,0,0,32,33,0,0,2,0,32,36,0,
     31,216,0,0,0,22
830 DATA 8,0,32,16,0,64,15,255,128,0,0,0,0 [4966]
,16,0,0,32,0,0,0,0,0,32,0,0,16,0,0,
     15,192,0,0,32,22
840 DATA 0,0,32,0,0,64,0,0,128,0,1,0,2,2,0 [3619]
,1,29,192,2,0,32,0,0,0,0,0,32,0,0,0,
     0,0,32,22
850 DATA 0,0,32,0,0,70,0,0,139,0,1,16,2,2, [3758]
33,1,29,222,2,0,32,0,0,0,0,0,32,0,0,
     0,0,0,32,22
860 DATA 8,0,32,16,0,64,15,255,128,0,0,0,0 [4393]
,32,0,0,80,0,0,136,0,1,4,0,2,3,128,0,
     0,64,0,0,32,22
870 DATA 0,0,32,0,0,0,0,0,32,7,254,64,8,1, [4226]
128,16,0,64,0,2,32,16,4,0,15,248,32,
     0,0,0,0,0,32,22
880 DATA 0,16,0,0,32,0,0,15,224,0,16,0,0,3 [5418]
2,0,0,16,0,0,15,224,0,16,0,0,32,0,0,
     16,0,0,15,224,22
890 DATA 0,16,0,0,32,0,0,16,0,0,15,224,0,1 [4233]
6,0,0,32,0,0,0,0,0,32,0,0,16,0,0,15,
     224,0,0,0,22
900 DATA 0,0,32,0,15,128,0,16,64,0,32,32,0 [4242]
,0,0,0,32,32,0,0,0,0,32,32,0,16,64,0,
     47,128,0,192,0,22
910 DATA 0,32,1,0,0,2,0,63,252,0,0,0,0,32, [4677]
32,0,0,0,0,32,32,0,0,0,0,32,32,0,16,
     64,0,15,128,22
920 DATA 0,15,128,0,16,64,0,32,32,0,0,0,0, [4831]
32,32,0,0,0,0,32,32,0,0,0,0,63,255,0,
     0,0,0,32,32,22
930 DATA 0,16,0,0,32,0,0,31,224,0,0,0,0,8, [4248]
0,0,16,0,0,32,0,0,24,0,0,4,0,0,0,0,0,
     4,0,22
940 DATA 0,0,128,0,1,0,0,2,0,0,4,0,0,8,0,0 [4051]
,16,32,0,32,0,0,24,32,0,4,64,0,3,128,
     0,0,0,22
950 DATA 0,32,32,0,0,64,0,32,128,31,223,0, [5133]
0,32,128,0,0,64,0,32,32,0,0,0,0,32,
     32,0,0,0,0,0,32,22
960 DATA 0,16,0,0,32,0,0,31,128,0,0,64,0,0 [3462]
,32,0,0,0,0,0,32,0,31,192,0,0,32,0,0,
     0,0,0,32,22
970 DATA 0,16,0,0,32,0,0,31,192,0,0,32,0,0 [4451]
,64,0,0,128,0,1,0,0,2,0,0,4,0,0,56,0,
     0,0,0,22
980 DATA 0,16,0,0,32,0,0,31,192,0,0,32,0,0 [5101]
,64,0,31,128,0,0,64,0,0,32,0,0,64,0,
     31,128,0,32,0,22
990 DATA 0,16,0,0,32,224,0,0,0,0,32,32,0,2 [3303]
9,192,0,2,0,0,61,128,0,0,64,0,32,32,
     0,24,0,0,0,32,22
1000 DATA 0,0,32,0,0,66,0,63,129,0,0,64,0, [3871]
0,33,0,0,0,0,0,33,0,0,2,0,63,252,0,0,
     0,0,0,0,22
1010 DATA 0,0,0,0,8,32,0,16,64,0,32,128,0, [4408]
1,0,0,34,32,0,4,0,0,40,32,0,16,64,0,
     32,128,0,0,0,99
1020 PRINT #8,; CHR$(27) + "%" + CHR$(1);   [1660]
1030 PRINT #8,; CHR$(7);                     [923]
1040 LOCATE 12,14:PRINT "Download durchgef [2159]
uehrt"
1050 LOCATE 12,15:PRINT "Sie koennen jetzt [6786]
 Ihre Textverarbeitungsprogramm starten"
1060 NEW                                     [318]
```

Listing DOWNLOAD

Rolle vor- und rückwärts

Exquisiter Bildaufbau

Zerknitterte, spröde Pergamente, die behutsam zusammengerollt verborgene Geheimnisse enthalten, kennt jeder. Will man sich der darauf stehenden Schriften kundig machen, ist es stets ein Akt größter Spannung, wenn sich jene "entrollen". Warum nicht eine solche Enthüllung auch auf dem CPC vollbringen?

Das Programm "Abrollen" schafft genau diesen Effekt. Ein beliebiges Bild wird vom Datenträger in den Speicher des CPC geladen und auf dem Bildschirm aufgerollt. Voraussetzung zum erfolgreichen Gebrauch des Utilities, beispielsweise in eigenen Programmen, ist, daß der Bildschirminhalt mit

SAVE "SCREEN",b,&c000,&3FFF

abgespeichert wurde. Die so gesicherte Datei muß mit

MEMORY &5FFF : LOAD "SCREEN",&6000

geladen werden. Aktiviert wird das visuelle Spektakel mit einem Call-Befehl. Die Routine, die an Adresse &A000 abgelegt wird, muß mit folgenden Parametern aufgerufen werden.

CALL &A000,Rollenbreite,Modus,Geschwindigkeit

Rollenbreite:
Hiermit wird die Breite der Rolle festgelegt, die auf dem Bildschirm abgerollt wird. Der Wert kann zwischen 2 und 100 liegen; empfohlen wird eine Einstellung, die zwischen 10 und 30 zu finden ist.

Modus:
Dieser Parameter entscheidet die Richtung, in der der Bildschirm geöffnet wird. Wählen Sie diesen Wert 0, so rollt er das Bild von oben nach unten, bei 1 selbstredend umgekehrt.

Geschwindigkeit:
Wenn dieser Parameter ungleich 0 ist, so wird die Aktion etwas beschleunigt; der Strahlenrücklauf findet dann keine Berücksichtigung.

Zwei Listings gilt es abzutippen, um in den Genuß dieser Sehenswürdigkeit zu kommen.

Das Listing "Abrollen.dat" sollte zuerst in den Computer gebracht werden. Starten Sie dieses. Es erzeugt die Binärdatei "Abrollen.bin". Danach können Sie das Demoprogramm laufen lassen. Denken Sie bitte daran, daß das Demoprogramm natürlich ein nettes Bild benötigt, um Ihnen eine entsprechende Präsentation zu bieten.

(Frank Schimmel/jf)

für 464-664-6128

```
100 '                                  [117]
110 ' Demo-Programm zu ABROLLEN        [2867]
120 '                                  [117]
130 ' von Frank Schimmel 1989          [1945]
140 '                                  [117]
150 '                                  [117]
160 MODE 1                             [506]
170 ' Farben fuer DATABOX Titelscreen  [1880]
180 BORDER 0                           [1008]
190 INK 0,13:INK 1,0:INK 2,9:INK 3,20  [1919]
200 BORDER 20                          [957]
210 MEMORY &5FFF                       [102]
220 '                                  [117]
230 ' Routine Abrollen laden           [2339]
240 '                                  [117]
250 LOAD"abrollen.bin"                 [1509]
260 '                                  [117]
270 ' Screen laden                     [974]
280 '                                  [117]
290 LOAD"screen",&6000                 [1193]
300 '                                  [117]
310 ' Start des "DemoABROLLEN"         [1813]
320 '                                  [117]
330 modus=-1                           [939]
340 CLS                                [91]
350 FOR i=1 TO 5                       [449]
360 NEXT                               [350]
370 READ breite                        [698]
380 IF breite=0 THEN RUN 330           [341]
390 modus=modus*(-1)-1                 [1044]
400 fast=0                             [85]
410 CALL &A000,breite,ABS(modus),fast  [1384]
420 fast=1                             [88]
430 WHILE INKEY$="":WEND               [1607]
440 CLS                                [91]
450 FOR i=1 TO 5                       [449]
460 NEXT                               [350]
470 CALL &A000,breite,ABS(modus),fast  [1384]
480 PRINT CHR$(7);                     [1175]
490 WHILE INKEY$<>"":WEND              [1786]
500 WHILE INKEY$="":WEND               [1607]
510 GOTO 340                           [464]
520 DATA 18,5,30,20,10,50,100,0        [879]
```

Listing ROLLE

```
10 ad=&A000:sz=7:ln=300:lnstp=10               [2492]
20 ON ERROR GOTO 60                            [1518]
30 su=0:READ a$:cs=VAL("&"+a$):FOR i=ad TO     [7449]
 ad+sz:READ a$:a=VAL("&"+a$):POKE i,a:su=s
u+a:NEXT:IF cs<>su THEN PRINT"DATA ERROR i
n"+STR$(ln):END
40 ad=ad+sz+1:ln=ln+lnstp:GOTO 30              [3902]
50 SAVE"abrollen.bin",b,&A000,&DC:END          [1521]
60 IF ERR=4 THEN RESUME 50                     [2203]
70 ON ERROR GOTO 0                             [1381]
300 DATA 03FB,FE,03,C0,DD,7E,00,B7,28          [1453]
310 DATA 034A,0B,21,D6,A0,97,77,23,77          [1545]
320 DATA 028A,23,77,18,0B,21,D6,A0,36          [1356]
330 DATA 0332,CD,23,36,19,23,36,BD,DD          [1043]
340 DATA 027E,7E,02,B7,20,0F,21,57,A0          [1526]
350 DATA 0275,36,00,23,36,C0,21,65,A0          [1703]
360 DATA 01CF,36,26,18,0D,21,57,A0,36          [1487]
370 DATA 0334,80,23,36,FF,21,65,A0,36          [1648]
380 DATA 03C2,29,DD,7E,04,32,9D,A0,CB          [2067]
390 DATA 024D,27,3D,3D,32,D0,A0,06,04          [1763]
400 DATA 02D3,3C,10,FD,32,97,A0,21,00          [1913]
410 DATA 03AA,C0,11,DC,A0,06,C8,7D,12          [1470]
420 DATA 0273,7C,13,12,13,CD,26,BC,10          [2202]
430 DATA 0484,F5,21,6C,A2,E5,D1,13,97          [1037]
440 DATA 0363,77,01,50,00,ED,B0,DD,21          [1470]
450 DATA 04E4,DC,A0,FD,21,DC,A0,06,C8          [1583]
460 DATA 03D0,C5,DD,6E,00,DD,66,01,7C          [1996]
470 DATA 044B,D6,60,67,FD,5E,00,FD,56          [2018]
480 DATA 0217,01,01,50,00,ED,B0,06,22          [1569]
490 DATA 03C4,DD,23,10,FC,06,10,C5,DD          [1246]
500 DATA 0470,2B,DD,2B,FD,23,FD,23,FD          [1629]
510 DATA 0307,5E,00,FD,56,01,7B,B2,28          [1723]
520 DATA 0294,1B,01,50,00,DD,6E,00,DD          [1462]
530 DATA 0318,66,01,7C,B5,28,06,7C,D6          [585]
540 DATA 03A9,60,67,18,06,D5,E1,97,77          [1340]
550 DATA 0361,13,0B,ED,B0,C1,10,CF,06          [1337]
560 DATA 0313,1E,FD,2B,10,FC,C1,00,00          [2451]
570 DATA 017E,00,10,A5,C9,00,00,00,00          [1628]
```

Listing ROLLE

Aus zwei mach eins

Verknüpfen von Multiplan-Tabellen

Multiplan bietet die elegante Möglichkeit, Tabellen logisch in der Art zu verknüpfen, daß Parameter und Ergebnisse von einer Tabelle in die andere übertragen werden können. Näheres siehe Handbuch. Eine physikalische Verknüpfung von Tabellen ist grundsätzlich nicht möglich. Anders ausgedrückt, die Übernahme von Formeln aus einer oder mehreren Tabellen in eine andere ist nicht implementiert. Mit einem Trick (und einem Texteditor) läßt sich dieser Mangel (sofern er als solcher empfunden wird) beheben:

Die physikalisch zu verknüpfenden Tabellen werden als SYLK-Tabellen (ASCII-Dateien) abgespeichert und mittels eines Editors zu einer neuen Datei verknüpft (mit dem Blockladebefehl nacheinander einlesen). An den "Nahtstellen" der Teiltabellen sind mindestens die Satzkennzeichen für Dateiende und -beginn der Teiltabellen zu löschen (ID;... und E). Danach kann man mit Multiplan die neue (SYLK-)Tabelle lesen. Ein Beispiel soll die Vorgehensweise verdeutlichen:

Tabelle 1 und 2 sind mit Multiplan erstellt und sollen zu Tabelle 3 physikalisch verknüpft werden:

Tabelle 1

A:	10
B:	10
Summe:	20

ÜBERTRAGEN OPTIONEN Format: Normal (Symbolisch) Fremd Laufwerk/Inhaltsverzeichnis
ÜBERTRAGEN SPEICHERN Dateiname: tab1.slk

Tabelle 2

A:	10
B:	10
Produkt:	100

ÜBERTRAGEN OPTIONEN Format: Normal (Symbolisch) Fremd Laufwerk/Inhaltsverzeichnis
ÜBERTRAGEN SPEICHERN Dateiname: tab2.slk

Wichtig: Aus leicht nachvollziehbaren Gründen dürfen die Teiltabellen sich in den verwendeten Feldern nicht überschneiden!

Nachdem mit Multiplan die Teiltabellen als SYLK-Tabellen abgespeichert wurden, wird mit einem Editor eine Programm-Datei namens tab3.slk eröffnet, in die mit dem Blockladebefehl nacheinander die Teiltabellen tab1.slk und tab2.slk geladen werden:

```
ID;PMP                          "Beginn Tabelle 3" zugleich
                                "Beginn Tabelle 1"
F;DG0G10
B;Y7;X3
C;Y3;X2;K"Tabelle 1"
C;Y5;K"A:"
C;X3;K10
C;Y6;X2;K"b:"
C;X3;K10
C;Y7;X2;K"Summe:"
C;X3;E+R[-1]C+R[-2]C;K-
20
W;N1;A1 1
E                               "Ende Tabelle 1" :Löschen
ID;PMP                          "Beginn Tabelle 2":Löschen
F;DG0G10
B;Y7;X6
C;Y3;X5;K"Tabelle 2"
C;Y5;K"A:"
C;X6;K10
C;Y6;X5;K"B:"
C;X6;K10
C;Y7;X5;K"Produkt:"
C;X6;E+R[-1]C*R[-2]C;K-
100
W;N1;A1 1
E                               "Ende Tabelle 2" zugleich
                                "Ende Tabelle 3"
```

Nachdem auf diese Weise die Teiltabellen verknüpft wurden, können sie mit Multiplan als neue Tabelle geladen und weiterverarbeitet werden:

ÜBERTRAGEN OPTIONEN Format: Normal (Symbolisch) Fremd Laufwerk/Inhaltsverzeichnis
ÜBERTRAGEN LADEN Dateiname: tab3.slk

Tabelle 1		*Tabelle 2*	
A:	10	A:	10
b:	10	B:	10
Summe:	20	Produkt:	100

Die neue Tabelle sollte anschließend als normale Tabelle abgespeichert werden. Sie kann nun ohne weiteres von Multiplan weiterverarbeitet werden.

Kurzanleitung

1. Tabelle 1 und 2 in Multiplan einladen und als SYLK-Tabelle abspeichern.
2. Multiplan verlassen und einen Texteditor starten.
3. Tabelle 1 einladen und die letzte Zeile löschen.
4. Tabelle 2 hinzuladen, deren erste Zeile löschen und eventuelle Leerzeilen zwischen Tabelle 1 und 2 entfernen.
5. Unter neuem Namen abspeichern und in Multiplan einlesen.

(Reiner Dienlin/rs)



JUCA

Ein Adreßverwaltungs-Programm für dBase-II-Fans (3)

Im dritten und letzten Teil der Serie befassen wir uns mit dem Teilprogramm MV3000 (Drucken), der Schnittstelle von dBase II zu LocoScript 2.16 und der Nutzung des Programms LocoMail 2. Wenn Sie im Besitz der drei Databoxen sind, stehen Ihnen ein umfangreiches Programm für die Adreßverwaltung unter dBase II, ein Handbuch zu JUCA und einige Bonusprogramme zur Verfügung.

Komplettierung der JUCA Start-DISK

Kopieren Sie bitte zunächst die JUCA-Dateien mit der Endung .CMD und die Dateien MVMAIL.216 und MVBRIEF.216 von der Databox dieses Heftes auf die Seite 2, Gruppe 0, der DISK A und die Datei MVADRGRP.00 auf die Seite 1, Gruppe 0, der DISK A. Damit ist das Programm vollständig. Die Dateien auf DISK A und DISK B sollten mit den in Abbildung 3 (Teil 1) genannten Dateien übereinstimmen!

So arbeiten Sie mit dem J U C A Teilprogramm 'Drucken' (MV3000)

21. Im Hauptmenü wählen Sie [3] und im Drucken-Menü [1], um die Adressenliste auszudrucken. Dieses geschieht entsprechend den Eingaben, die Sie in den Feldern 'Gruppen Nr.' und 'Namen von:' sowie 'Namen bis:' und 'Interesse' vornehmen beziehungsweise – falls Sie alle Adressen gedruckt haben möchten – nach [RET] [RET] [RET]. Geben Sie jedoch zunächst Papier in den Drucker (der rechte Papierrand liegt bei '80' am Andrückbügel), nehmen Sie mit [EXIT] die Statuszeile weg, drücken Sie erst dann [j]. Mit der Schlußeingabe 'n' vor Beginn des Drucks können Sie einen Abbruch erreichen. Abbildung 8 zeigt die fertige Adressenliste. Wir waren der Meinung, daß die Übersichtlichkeit durch die Strichelung zwischen zwei Adressen erhöht wird.

22. Wollen Sie die Abkürzungen und zugehörigen Erläuterungen ausdrucken, so wählen Sie im Drucken-Menü die Option [2]. Nach den entsprechenden Eingaben beziehungsweise [RET] [RET] (Papier!) und einem [j] beginnt der Ausdruck. Die Programmierung ist so erfolgt, daß platzsparend in zwei Spalten gedruckt wird. Sie müssen also das Blatt, sobald eine Spalte gedruckt ist, ein zweites Mal durch den Drucker 'jagen'! Es ist wichtig, daß Sie das Papier immer an der gleichen Stelle in den Drucker einziehen (rechter Blattrand bei '80' auf dem Andrückbügel) und darauf achten, daß auch der obere Blattrand so wie beim Druck der ersten Spalte anliegt (zum Beispiel in Höhe der Gummirollen). Unterbleibt das, so werden die Spalten nicht in gleicher Höhe ausgedruckt! Abbildung 9 zeigt die Abkürzungliste. Nach dem Ausdruck der Listen kehrt das Programm in das Drucken-Menü zurück.

Der Spalten- und Seitenumbruch in der Datei MV3200 ist so interessant, daß hier der entscheidende Teil des Listings wiedergeben werden sollte:

```
STORE 1 TO zeile
STORE "1" TO spalte
SET FORMAT TO PRINT
DO WHILE .NOT EOF
IF .NOT EOF
IF zeile = 1 .AND. spalte = "1"
STORE seitenr + 1 TO seitenr
@ zeile,30 SAY "V E R Z E I C H N I S D E R A B
K Ü R Z U N G E N"
@ zeile,100 SAY "Seite : "
@ $,$+1 SAY seitenr USING '999'
@ zeile + 1,0 SAY d50 + d50 + d10 + d10 + d5
STORE 4 TO zeile
ENDIF
IF zeile = 1 .AND spalte = "r"
STORE 4 TO zeile
ENDIF
IF spalte ="1"
@ zeile,5 SAY ABKUERZ
@ zeile,11 SAY "="
@ zeile,13 SAY BEGRIFF
ELSE
@ zeile,62 SAY ABKUERZ
@ zeile,68 SAY "="
@ zeile,70 SAY BEGRIFF
ENDIF
STORE zeile + 2 TO zeile
IF zeile >= 127 .AND. spalte="1"
STORE 1 TO zeile
STORE "r" TO spalte
EJECT
ENDIF
IF zeile >= 127 .AND. spalte="r"
STORE 1 TO zeile
STORE "1" TO spalte
EJECT
ENDIF
CONTINUE
ENDIF
ENDDO
USE
EJECT
```

und so weiter.

23. Aus Menü 'Drucken' ist mit der Option [3] der JUCA-Programmteil zu starten, der für LocoScript 2.16 eine ASCII-Adressen-Kartei erstellt; die Musterdatei 'MVADRGRP.00' befindet sich bereits auf Seite 1, Gruppe 0, der DISK A. Nach Eingabe der 'Gruppen Nr.:' und der 'Namen von:' sowie 'Namen bis:' beziehungsweise einem dreimaligen 'Hipp Hipp Hurra' – Sor-

ry! - [RET] [RET] [RET] erscheinen die Hinweise 'Selektion läuft' und dann 'Zieldiskette liegt im Laufwerk:'. Geben Sie jetzt die DISK ins LW, auf der Sie die ASCII-Adressen-Liste später weiter verarbeiten wollen, in der Regel also eine Locoscript-Daten-Diskette.

Jetzt nehmen Sie einfach Seite 1 von DISK B und tippen den Buchstaben für das entsprechende LW ein ([B]). Beachten Sie auf jeden Fall den Hinweis am unteren Bildrand 'Kopiervorgang läuft'; die DISK darf während dieses Vorgangs nicht aus dem LW genommen werden! Nach der Aufforderung 'Legen Sie die alte Diskette wieder ins Laufwerk:' verfahren Sie entsprechend und melden Ihre Programm-DISK durch Eingabe des LW-Buchstabens ('A' beziehungsweise 'B') wieder an. Mit [0] kehren Sie zum Hauptmenü zurück.

Bei der Selektion der Daten verfährt das Programm so, daß es für die vorgesehenen Felder eine Zwischendatei auf LW M> anlegt:

```
COPY TO m:mv.temp FIELD name, vorname,
postltz, ort und strasse
```

N A M E PLZ O R T	V O R N A M E B E R U F S T R A B E	TEL. priv. TEL. dienst.	B E R A T I N T E R E S S E N	Seite : 1
Axelschweiß 3001 Hahnover 3	Adolfo Hochschulprofessor Prufungsweg 77	0918-92839192 0918-1111-234	JOY1 RAMER L3.5 LOCO2 DBASE WORDS SPIEL ASTRO	
Bollermann 8012 Minchen 123	Bertha Dipl.-Näherin Zwirnweg 15	011-999999	JOY2 LOCO1 DFERN RAPPO ASTRO JPRO1 SMALC DRLDB	
Drastenflosser 1234 Dässeldurf	Hubert Fischer Heringssteg 81	987-654321 987-101010	JOY1 LOCO2 WAWIS	
Beißbart 9000 Bärlin 73	Wilhelm-Ottokar Förster Am Wald 100	063120-13 063120-26	JOY2 LOCO2 DBASE DRBRA TURPA	
Haarfresser 8234 Blacknank 19	Albertstein Friseur Turpeegasse 81	0412-6704 0412-996633	JOY2 LASER PROWO SALON DRLOG	
Hundeknödel 2020 Hammelburg 1	Wadilav Nudelmacher Bellostreet 8	0042-8888 0043-12-1987	JOY1 RAMER SCHNI LOCO1 JUCA QBIL2 DFERN DRBRA TURPA	
Kleisterblatt 6412 Dordmunt 23	Justus Malermeister Kleiner Leimkamp 99 A	456-1289 456-1289	JOY2 MAR40 LOTTO KFZAB LOCO2 BASIC	
Wunderlich 5722 Freifurt/Mein	Clodhilde Hausmaklerin Gebührenplatz 6	0817-718293 0817-718239	JOY2 LOCO2 IMOFO DBASE SMALC	

Abbildung 8: Ausdruck der Adressen mit dem JUCA-Modul MV3100 (Drucken / Drucken Personen)

Nachdem die Ziel-DISK für die Selektionsdaten im LW liegt, greift JUCA auf diese Zwischendatei zurück, und kopiert die Daten zur Ziel-Datei 'MVADRGRP.00' und löscht die Zwischendatei dann wieder:

```
USE m:mv.temp
COPY TO &zieldatei SDF DELIMITED WITH ,
USE
DELETE FILE m:mv.temp
```

Die Einleitung der Programmzeile mit der Option SDF bewirkt, daß dBase die Zieldatei im Standard ASCII-Format anlegt, so daß sie von anderen Programmen wie LocoScript verstanden wird. Der dBase-Befehl 'DELIMITED WITH ,' hat zur Folge, daß die übertragenen Daten durch Kommata getrennt werden.

24. Vom Hauptmenü aus beenden Sie das Programm durch die Eingabe [0]. Es erscheint der Hinweis 'PACK durchführen? (j/n):'. Geben Sie [j] ein, so wird Ihr Auftrag prompt ausgeführt, das heißt, alle als gelöscht markierten Datensätze werden endgültig beseitigt. Je größer die Datenbänke sind, um so länger dauert dieser Vorgang! Beachten Sie den Hinweis 'PACK wird durchgeführt, bitte warten!'. Sobald

```
V E R Z E I C H N I S   D E R   A B K U R Z U N G E N                    Seite :    1
------------------------------------------------------------------------------------------
AAAAA = AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA      SPIEL = Spiele aller Art
ARCHE = Dateiverwaltung                           STOPP = Stop Press Desktop-Publishing
ASTRO = Astrologieprogramm (MUKRA)                SUCA2 = SuperCalc2 engl. Kalkulationsprogramm
BASIC = Programmiersprache Basic                  SUPT2 = Supertype II Zusatzschriften fur LOCO2
BIORH = Biorhythmus (MUKRA)                       TEST  = Abfrageprogramm mit Multiple Choice
CAKAS = Carat-Kasse (MicroMarket-Worms)           TURPA = Programmiersprache Turbo Pascal 3.0
COKAS = Comac Kasse (Einnahme-uberschußrechng.)   TUTUT = Turbo Tutor Einführung zu Turbo Pascal
DBASE = dBASE II (Version 2.41) (MARKT&TECHNIK)   VARD2 = VarDat II Benutzeroberfläche für DBASE
DBASS = dB-Ass: Benutzeroberfläche für DBASE      VIDEO = Videothekverwaltung
DBLAG = dB Lager Lagerverwaltung mit dBASE II     VOKTR = Vokabeltrainer
DFERN = Datenfernübertragung                      WAWIS = Warenwirtschaftssystem (BYTE ME)
DRGRA = Grafikprogramm DR GRAPH (SCHNEIDER DATA)  WORDS = Wordstar Textverarbeitg (Markt&Technik)
```

Abbildung 9: Ausdruck der Abkürzungen mit dem JUCA-Modul MV3200 (Drucken / Drucken Abkürzungen)

dieser Vorgang beendet ist, werden vollautomatisch die Tastatur und der Drucker wieder zurückgesetzt, das heißt die Standard-Einstellung wird wiederhergestellt. Danach erscheint das CP/M - Prompt M>.

25. Sie sollten nun DISK B aus dem LW nehmen und gegen die DISK C ersetzen, um anschließend die Schritte 03. bis .24 nochmals zu üben. Dabei geschieht das unter Schritt .05 Besprochene, das heißt, bei der Frage nach dem Anlegen der Datenbankdateien müssen Sie diesmal mit [j] antworten! Sie können jetzt mit der Eingabe 'echter' Daten beginnen!

Wie Sie die Adreß-Kartei unter LocoMail weiterverarbeiten

26. Starten Sie LocoScript 2.16 und legen Sie die Diskette ein, auf der Sie die ASCII-Datei 'MVADRGRP.00' angelegt haben (DISK A oder B, Seite 1). Kopieren Sie die Datei in die Gruppe 0 von LW M>.

27. Kopieren Sie von der DISK A (Seite 2) ebenfalls die Dateien MVMAIL .216 und MVBRIEF.216 in die Gruppe 0 von LW M>.

28. Rufen Sie die Datei MVMAIL.216 mit [B] [ENTER] [ENTER] auf. Unter der Textbearbeitung sehen Sie die gleichen Feldbezeichnungen, wie sie unter JUCA benutzt werden: Name, Vorname, PLZ, Ort, Straße. Setzen Sie den Cursor an den Beginn von Seite 2 und rufen Sie über [f1] die Option 'Text einfügen' auf. Wenn die Diskverwaltung erscheint, stellen Sie den Cursor auf die Datei 'MVADRGRP.00' und drücken zweimal [ENTER]. Die ASCII-Datei wird nun in die LocoMail-Schablone eingelesen (Abbildung 10). Ist das geschehen, so speichern Sie die Datei unter der Bezeichnung 'LOCOMAIL.ADR' auf der DISK ab.

29. Rufen Sie nun die Datei MVBRIEF.216 mit [B] [ENTER] [ENTER] auf und sehen Sie sich den vorbereiteten Inhalt an. In diesem Brief sind die Feldbezeichnungen eingebaut, die Sie schon in der Datei MVMAIL. 216 beziehungsweise LOCOMAIL. ADR vorgefunden haben. Verlassen Sie die Datei und kehren Sie zur Diskverwaltung zurück.

30. Setzten Sie nun den Cursor auf die Datei MVBRIEF.216 und rufen Sie den Misch-Modus mit [M] auf. Sie werden jetzt aufgefordert, den Cursor auf die Daten-Datei zu stellen, in unserem Fall also LOCOMAIL.ADR. Wenn Sie [ENTER] drücken, wird der Brief aufgerufen, und der Mischvorgang läuft ab. Sollten Sie mit der Vorgehensweise Probleme haben, so empfehle ich Ihnen das Studium des neuen LocoMail-Handbuchs beziehungsweise den Artikel in der PCAI 10/88 (S. 83ff.: "LocoMail von Kopf bis Fuß") und die zugehörende JOYCE Databox 10/88.

Möchten Sie über die Daten Name, Vorname, PLZ, Ort und Straße hinaus weitere Felder aus Ihrer Datenbank übertragen, so müssen Sie die Anweisungen in der Datei MV3300 um die gewünschten Felder ergänzen. Ebenso dürfen Sie nicht vergessen, die neuen Feldbezeichnung in den LocoMail-Dateien einzufügen!

Schlußbemerkungen

Das Programm JUCA ist mit großer Sorgfalt erstellt worden. Sie sollten daher nicht versäumen, das JUCA-Listing auszudrucken und es durchzuarbeiten.

Wenn Sie sich bislang über das endlos lange Copyright von dBase II geärgert haben, so möchten wir Sie auf den Artikel 'dBase II Patch' (PCI 12/89) hinweisen.

Hier wird beschrieben, wie die Kürzung des Copyright zu bewerkstelligen ist. Nach diesem Patch macht der Start von dBase II beziehungsweise JUCA noch mehr Spaß. Nebenbei wird auch für die Eindeutschung von "Gesamt" gesorgt. (Hinweis: Die Firma acw-Soft, Breite Str. 16, 53 Bonn 1, hat in ihrer Anleitung zur dBase-Benutzeroberfläche dB-ASSI auf weitere Patches hingewiesen).

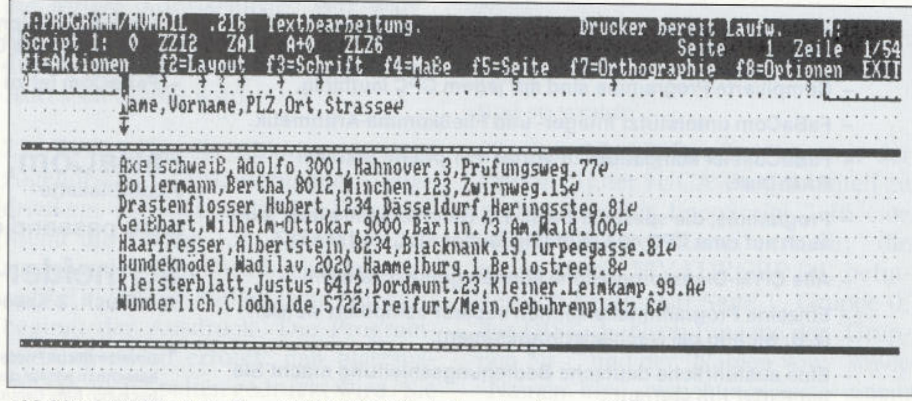
Weitere Verbesserungen können Sie erreichen, wenn Sie in die SUB-Dateien Programme wie SPEEDER (JOYCE SH 1), SPOOL.COM (JOYCE SH 3 und PCAI 10/88) und DIR.COM (Systemdisk 2) einbauen. Gerade die Nutzung von SPOOL.COM ist empfehlenswert, weil Sie dann frühzeitig unter CP/M PLUS, dBase II oder JUCA weiterarbeiten können, obwohl der von JUCA aus gestartete Druckvorgang noch läuft!

Wir würden uns freuen, wenn Sie uns über Ergänzungen und Verbesserungen bei JUCA informieren würden. Zum Schluß möchten wir Sie animieren, auch Ihr dBase-Superprogramm den Lesern der PC-AMSTRAD INT. zur Verfügung zu stellen.

Hinweise zur DATABOX dieses Heftes

Auf der DATABOX finden Sie alle Dateien für das Teilprogramm MV3000 (Drucken), die im Text erwähnten LocoMail-Dateien und eine Schablone zum Einlesen der veröffentlichten Texte zwecks Handbucherstellung.

(Wolfgang Ertel/Detlef Gehring/rs)



```
A:PROGRAMM/MVMAIL  .216  Textbearbeitung.                  Drucker bereit Laufw.   M:
Script 1:  0  ZZ12  ZA1  A+0  ZLZ6                                Seite   1  Zeile  1/54
f1=Aktionen  f2=Layout  f3=Schrift  f4=Maße  f5=Seite  f7=Orthographie  f8=Optionen  EXIT

Name,Vorname,PLZ,Ort,Strasse

Axelschweiß,Adolfo,3001,Hahnover.3,Prüfungsweg.77
Bollermann,Bertha,8012,Minchen.123,Zwirnweg.15
Drastenflosser,Hubert,1234,Dässeldurf,Heringssteg.81
Geißbart,Wilhelm-Ottokar,9000,Bärlin.73,Am.Wald.100
Haarfresser,Albertstein,8234,Blacknank.19,Turpeegasse.81
Hundeknödel,Wadilav,2020,Hammelburg.1,Bellostreet.8
Kleisterblatt,Justus,6412,Dordmunt.23,Kleiner.Leimkamp.99.A
Wunderlich,Clodhilde,5722,Freifurt/Mein,Gebührenplatz.6
```

Abbildung 10: Die Schablone MVMAIL.216 mit der eingefügten ASCII-Datei MVADRGRP.00

VDE

Ein Public-Domain-Editor für ASCII-Texte

Eine Schwachstelle des Amstrad PCW ist, daß standardmäßig unter CP/M nur zwei ASCII-Editoren zur Verfügung stehen, die für die Eingabe längerer Texte höchst umständlich beziehungsweise schlicht ungeeignet sind. Wer sich daher nicht ein Textverarbeitungsprogramm auch für CP/M gekauft hat, war bisher gezwungen, zwischen seinen CP/M-Anwendungen und LocoScript hin- und herzuwechseln.

"Mensch Meyer, daß das nicht besser geht!", mag bisher manch RPED- oder ED-geplagte PCW-User gedacht haben. Und E. MEYER aus Washington hat tatsächlich die Lösung: VDE - Video Display Editor (V 2.31) heißt sein nur 10 kBytes kleiner, aber ungemein flinker und vielseitiger ASCII-Editor für CP/M-Computer. Zusammen mit einem Installationsprogramm läßt sich dieser auf jeder CP/M-Maschine installieren und stellt daher auch für PCW- und CPC-Besitzer eine interessante Alternative zu anderen Editorprogrammen dar. Zumal VDE nur 10 kByte klein ist und Steuercodes verwendet, die stark an Wordstar erinnern.

Was 10 kByte können, ...

... kann manchen in Staunen versetzen. Folgende Operationen stehen bei diesem bildschirmorientierten Editor zur Verfügung:

* Cursorbewegungen über den gesamten Bildschirm; zum Anfang / Ende der Seite, der Zeile, der Datei; zu einem Blockmarker; zum vorigen / nächsten Zeichen oder Wort und so weiter

* Löschen von Zeichen, Wörtern, Zeilen und Blöcken. Und, man höre und staune, eine UNDELETE-Funktion, mit der bis zu 2 kByte an gelöschtem Text wieder zurückgeholt werden können!

* Blockoperationen: Kopieren eines Blocks in und aus einem Bufferspeicher, auf und von Diskette.

* Suche und suche/ersetze.

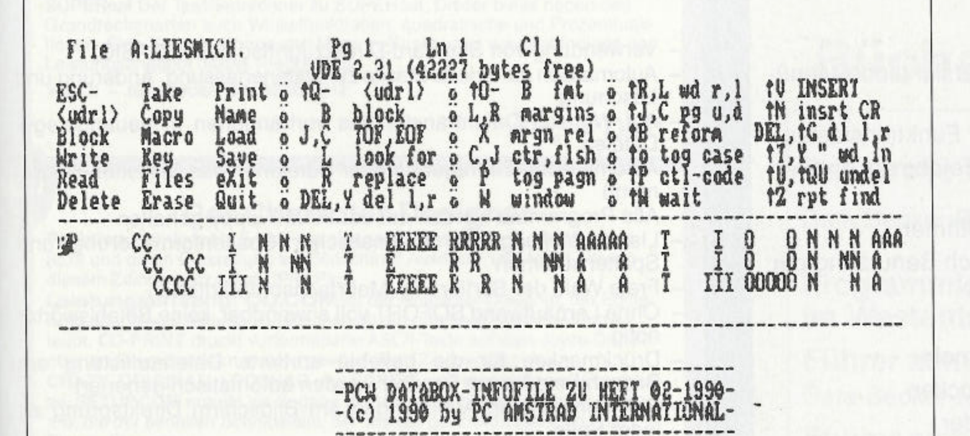
* Layout: linker / rechter Rand beliebig oft pro Text wählbar; Zentrieren; rechtsbündiges Plazieren; neu Formatieren.

* Ausdrucken mit und ohne Kopfleiste und Seitennumerierung.

* Einfügen / Überschreiben ist wählbar; zwischen echtem Tabulator oder entsprechender Anzahl von Leerzeichen kann ebenfalls gewählt werden. Mit Tastendruck können Großbuchstaben in kleine verwandelt werden und umgekehrt.

* Von VDE aus können Dateien gelöscht, umbenannt oder geladen werden und das Directory kann gezeigt werden.

* Zehn Tasten können mit Macros belegt werden. Der automatische Wortumbruch kann zugeschaltet werden und noch einiges mehr!



```
File  A:LIESMICH.          Pg 1     Ln 1      Cl 1
                     VDE 2.31 (42227 bytes free)
ESC-    Take   Print ö ↑Q-   (udrl)   ö ↑O-  B  fmt     ö ↑R,L wd r,l   ↑V INSERT
(udrl)  Copy   Name  ö   B  block     ö L,R  margins    ö ↑J,C pg u,d   ↑N insrt CR
Block   Macro  Load  ö J,C  TOF,EOF   ö  X   mrgn rel   ö ↑B reform    DEL,↑G dl l,r
Write   Key    Save  ö   L  look for  ö C,J  ctr,flsh   ö ↑ö tog case   ↑T,Y " wd,ln
Read    Files  eXit  ö   R  replace   ö  P   tog pagn   ö ↑P ctl-code   ↑U,↑QU undel
Delete  Erase  Quit  ö DEL,Y del l,r  ö  W   window     ö ↑W wait       ↑Z rpt find
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------

                         --------------------------------
                         -PCW DATABOX-INFOFILE ZU HEFT 02-1990-
                         -(c) 1990 by PC AMSTRAD INTERNATIONAL-
                         --------------------------------

Auf dieser Databoxdiskette sind entsprechend den Artikeln folgende Dateien auf d
Diskette enthalten:

    Artikel                          Zugehörige Dateien
```

Abb1.: Über EXIT-'?' wird dieses Menü auf dem Bildschirm dargestellt

Ein Blick durchs Fenster ...

... ist auch bei VDE möglich. Hier heißt das dann WINDOWING und sieht etwas anders aus, als es LocoScript-verwöhnte PCW-User nun erwarten. VDE erlaubt eine Teilung des Bildschirms in zwei Hälften. In einer werden die elf Zeilen ab der Position des Cursors gezeigt, während die andere wie gewohnt weiterfunktioniert. Damit lassen sich Textstellen innerhalb einer Datei, aber auch von zwei verschiedenen Dateien vergleichen. Hier zeigt sich allerdings eine Schwäche, da diese VDE-Funktion nur auf 24-Zeilen-Bildschirmen funktioniert. In der Praxis hat es sich aber gezeigt, daß es ausreicht, wenn beim Installieren ein 24-Zeilen-Bildschirm angegeben wurde. Keinesfalls notwendig ist ein Umschalten des JOYCE-Bildschirms auf 24 Zeilen x 80 Zeichen. Ebenfalls als Fenster in Form einer Kopfleiste läßt sich jederzeit ein Hilfsmenü einblenden, welches die wichtigsten Funktionen und Befehle von VDE zeigt.

VDE und die PCW-Tastatur

Eine Tastatur, die so auf Textverarbeitung ausgelegt ist wie die des PCW muß natürlich genutzt werden. Ein Problem ist dabei, daß natürlich die Standardbelegung von CP/M auch für andere Programme erhalten bleiben muß. Die dafür erstellte Tastaturbelegung VDE-PCW.KEY nimmt darauf Rücksicht und auch auf die Gewohnheiten, die sich aus der Anwendung von LocoScript ergeben. Sie stellt die Textverarbeitungstasten weitgehend unter VDE zur Verfügung, so daß nicht eine Menge neuer Steuercodes erlernt werden müssen. Zur leichteren Orientierung wurden auch ausführliche Kommentare in VDE-PCW.KEY aufgenommen. Der erste Text der mit VDE auf PCW üblicherweise dann geschrieben wird, lautet meist: VDE ! ED ade!

Das VDE - Programmpaket gibt es bei ADVANTAGE, 56 BATH ROAD, CHELTENHAM GL53 7HJ, Great Britain um 6 Pfund und mit VDE-PCW.KEY bei der JOYCE USER GROUP in Styria, Postfach 96, A - 8041 Graz um sS 200,–

(Herbert Moschitz/rs)

Von Assembler nach BASIC

Ein DATA-Lader für HEX-Dateien

Wer als BASIC-Programmierer auf Maschinen-Routinen nicht verzichten kann, steht immer wieder vor dem Problem, sein Maschinenprogramm in eine für BASIC verständliche Form zu bringen. Da MALLARD-BASIC weder Maschinencode noch Assembler-Quelltext direkt verarbeiten kann, muß die benötigte Maschinenroutine mittels POKE-Kommandos in den Speicher geladen werden.

Dies geschieht im allgemeinen über einen sogenannten DATA-Lader, in dem der Maschinencode in DATA-Zeilen (das sind Programmzeilen, die mit dem BASIC-Kommando 'DATA' beginnen) in dezimaler oder hexadezimaler Form definiert ist, und der meist als eigenständige Routine von einem BASIC-Programm aufgerufen werden kann (vgl. PC Amstrad 9/89, Programm Joyprint).

Nun sind professionelle Assembler-Programme aber normalerweise nicht in der Lage, einen solchen DATA-Lader zu generieren, das heißt, direkt aus dem Quelltext des Programms die benötigten DATA-Zeilen zu erzeugen. Der Quelltext eines Maschinenprogramms muß also zunächst in Maschinencode übersetzt werden (man nennt diesen Vorgang assemblieren), und dieser Maschinencode wird dann wieder in einen DATA-Lader umgewandelt, damit BASIC ihn verarbeiten kann.

Allerdings wurden mittlerweile auch schon für den JOYCE (PCW 8256/8512) Assembler-Programme veröffentlicht, die DATA-Zeilen erzeugen können, aber da diese Programme selbst in BASIC geschrieben wurden, lassen sie, was die Verarbeitungsgeschwindigkeit angeht, einiges zu wünschen übrig.

Gerade wenn man längere Maschinenprogramme zu übersetzen hat, wünscht man sich jedoch etwas mehr 'Beschleunigung'.

Umleitung

Die professionellen Assembler-Programme, die direkt unter CP/M arbeiten (z.B. MAC.COM oder M80.COM) sind natürlich, da sie auf Maschinenebene laufen, um einiges schneller. Sie können allerdings, wie schon gesagt, keine DATA-Zeilen erzeugen. Wollen wir nun, um die höhere Geschwindigkeit zu nutzen, trotzdem an unseren DATA-Lader kommen, müssen wir einen kleinen Umweg machen.

Umweg

Professionelle Assembler-Programme bieten gewöhnlich die Möglichkeit, HEX-Dateien zu erzeugen, das sind Dateien, in denen der Maschinencode in hexadezimaler Form mit normalen ASCII-Zeichen (dem sog. Intel-Hexadezimal-Format) abgespeichert wird. Abbildung 1 zeigt einige Zeilen einer solchen Datei.

Diese Dateien können unter anderem auch von dem CP/M-Dienstprogramm HEXCOM verarbeitet werden, um daraus 'normale' COM-Dateien zu erzeugen. Nebenbei haben diese Dateien aber auch noch den Vorteil, daß hier schon alle Informationen enthalten sind, die auch für den DATA-Lader benötigt werden. Neben den Adressen, unter denen das Programm später abgespeichert werden soll, sind alle Bytes des Maschinencodes schon so vorhanden (in ASCII-Form), wie sie auch im DATA-Lader auftreten, es müssen nur noch Kommata zur Trennung der Bytes eingefügt werden.

```
:1CD0000022A1D01193D0010A00EDB02A9BD0E9ED5BA1D02193D0010800EDB0C93B
:1CD01C00CD3AD00142D0182C013AD0CD50D01816ED4B95D0ED5B97D0CD5AFCAA8B
:1CD0380015C92A99D011A3D0180721A3D0ED5B97D0ED4B95D0EDB0C9CD5AFC401F
:1CD0540017C90163D018F5016BD0CD50D018AC3A93D02A99D077C92A99D07E32FF
```

Ausschnitt aus einer Datei im INTEL-HEX-Format. Hier sind schon alle Informationen vorhanden, die für den DATA-Lader benötigt werden.

Ein Programm muß her

Was liegt also näher, als ein kleines Programm zu schreiben, das diese Kommata einfügt und das Ganze in einen DATA-Lader verwandelt. Nun mit HEXDAT (analog zu HEXCOM) ist Ihnen diese Arbeit schon abgenommen. Sie können fortan beliebige HEX-Dateien in DATA-Lader umwandeln, die dann in Ihre BASIC-Programme integriert werden können.

Mit HEXDAT lassen sich auch mehrere Maschinenprogramm-Module miteinander 'verbinden', die dabei erzeugten DATA-Lader werden durch GOTO-Kommandos hintereinandergehängt und verbunden.

Komfort muß sein

Die Bedienung des Programms ist einfach. Es müssen nur zwei Parameter eingegeben werden: Der Dateiname der HEX-Datei und die Zeilennummer, bei der der DATA-Lader beginnen soll. Um das Ganze noch mehr zu vereinfachen, werden die auf dem ausgewählten Laufwerk vorhandenen HEX-Dateien angezeigt und die erste Datei als Voreinstellung benutzt. Die Zeilennummer wird vom Programm auf 50000 voreingestellt, so daß im günstigsten Fall ein zweimaliger Druck auf die RETURN-Taste genügt, um die Umwandlung von 'HEX' nach 'DATA' vorzunehmen.

Der erzeugte DATA-Lader wird unter gleichem Dateinamen, jedoch mit dem Typenbezeichner '.DAT' auf dem ausgewählten Laufwerk abgespeichert. Er kann dann direkt unter BASIC mit

RUN "Datname.DAT"

aufgerufen ('Datname' ist vom Namen Ihrer HEX-Datei abhängig) oder als Unterroutine in Ihr BASIC-Programm aufgenommen werden. Der Aufruf innerhalb Ihres Programms erfolgt dann mit

GOSUB Zeilennummer

wobei 'Zeilennummer' die für den DATA-Lader angegebene Startzeile sein muß.

(Norbert Finke/rs)

```
LISTING >HEXDAT  <, REMARK = >'<.

<11> 1000 '*------------------------------------------*
<74> 1010 '*.............. HEXDAT.BAS................ *
<56> 1020 '*... erzeugt aus einer HEX-Datei einen.... *
<90> 1030 '*... DATA-Lader für die Einbindung von.... *
<62> 1040 '*. Maschinen-Routinen in BASIC-Programme.. *
<57> 1050 '*...... written 1989 by NoFi............. *
<29> 1060 '*------------------------------------------*
<42> 1070 DIM dline$(1000),adr(20):lnbr!=50000!
<38> 1080 DEFINT a-z:WIDTH 255:ON ERROR GOTO 2110
< 8> 1090 GOSUB 1170:....... 'Bildschirm initialisieren
<47> 1100 GOSUB 1350:................ 'Parameter holen
<27> 1110 GOSUB 1510:..... 'Hexfile lesen und umwandeln
<23> 1120 PRINT cl$:GOSUB 1970:.... 'Datafile schreiben
<77> 1130 END
<25> 1140 '*------------------------------------------*
<52> 1150 '*......... INITIALISIERUNG................ *
<31> 1160 '*------------------------------------------*
<36> 1170 es$=CHR$(27):cl$=es$+"E"+es$+"H":PRINT cl$;
<47> 1180 iv$=es$+"p":nm$=es$+"q":ceol$=es$+"K"
<90> 1190 gcur$=es$+"j":pcur$=es$+"k"
<38> 1200 DEF FNat$(x,y)=es$+"Y"+CHR$(y+32)+CHR$(x+32)
< 6> 1210 DEF FNnbr$(i)=MID$(STR$(lnbr!+i),2)+" "
<48> 1220 DEF FNlnbr$(i)=">"+iv$+" "+DEC$(lnbr!+i,"####
     ####")+" "+nm$+"<"
<69> 1230 PRINT CHR$(150)STRING$(23,154)CHR$(156)
<89> 1240 PRINT CHR$(149)". "iv$" >> H E X D A T << "nm
     $". "CHR$(149)
<34> 1250 PRINT CHR$(147)STRING$(23,154)CHR$(153)FNat$(
     0,5);
<92> 1260 hfile$=FIND$("*.HEX"):PRINT STRING$(90,"=")
<63> 1270 IF hfile$<>""THEN FILES "*.HEX":GOTO 1300
<38> 1280 PRINT"Keine Datei des Typs "iv$" .HEX "nm$" g
     efunden!"
<17> 1290 hfile$=SPACE$(8)
<59> 1300 PRINT:PRINT STRING$(90,"="):PRINT:PRINT
<91> 1310 PRINT gcur$:RETURN
<23> 1320 '*------------------------------------------*
<60> 1330 '*.......... EINGABEN HOLEN............... *
<29> 1340 '*------------------------------------------*
<19> 1350 PRINT FNat$(0,30)"(Eingabe "iv$" END "nm$" =>
     Programm beenden)";
<54> 1360 PRINT pcur$ceol$"Dateiname.. >"iv$" "LEFT$(hf
     ile$,8)" "nm$"<?";
<57> 1370 INPUT;" ",a$
<27> 1380 IF UPPER$(a$)="END"THEN 2320
<49> 1390 IF a$=""THEN a$=hfile$
<24> 1400 IF FIND$(a$)=""THEN ERROR 53
<41> 1410 hfile$=a$:dfile$=MID$(a$,1,INSTR(a$,"."))+"DA
     T"
<60> 1420 PRINT:PRINT:PRINT gcur$FNat$(0,30)SPC(35);
<61> 1430 PRINT pcur$ceol$"Startzeile. "FNlnbr$(0)"?";
<50> 1440 INPUT;" ",a$
<91> 1450 IF VAL(a$)=0 THEN 1470
<84> 1460 IF VAL(a$)<65535! THEN lnbr!=VAL(a$)ELSE ERRO
     R 6
<78> 1470 PRINT:PRINT:PRINT gcur$:RETURN
<43> 1480 '*------------------------------------------*
<17> 1490 '*.............. HEXFILE LESEN.......... *
<21> 1500 '*------------------------------------------*
<70> 1510 a=0:byte=0:d=4:s=0:tadr=0
<31> 1520 dline$(d)=FNnbr$(d)+"DATA "
<50> 1530 OPEN"I",#1,hfile$
<66> 1540 WHILE NOT EOF(1)
<69> 1550 INPUT #1,hline$
<58> 1560 IF LEFT$(hline$,1)<>":"THEN ERROR 5
<12> 1570 GOSUB 1640:GOSUB 1740:..... 'Zeile umwandeln
<45> 1580 WEND
<18> 1590 CLOSE
<90> 1600 RETURN
<26> 1610 '*------------------------------------------*
<13> 1620 '*............ ADRESSE HOLEN.............. *
<32> 1630 '*------------------------------------------*
<56> 1640 ta=VAL("&H"+MID$(hline$,4,4))
<57> 1650 IF tadr+byte=ta OR ta=0 THEN RETURN
<19> 1660 tadr=ta:adr(a)=tadr:a=a+1
< 2> 1670 IF a=1 THEN RETURN ELSE a=a-2
```

Listing HEXDAT

```
<81> 1680 GOSUB 1890:s=d:d=d+4:byte=0:a=a+2
<54> 1690 dline$(d)=FNnbr$(d)+"DATA "
<92> 1700 RETURN
<28> 1710 '*------------------------------------------*
<35> 1720 '*............ ZEILE UMWANDELN............. *
<34> 1730 '*------------------------------------------*
<41> 1740 hl=VAL("&H"+MID$(hline$,2,2))*2
<39> 1750 byte=byte+hl/2:hline$=MID$(hline$,10,hl)
<23> 1760 FOR i=1 TO hl STEP 2
<46> 1770 dline$(d)=dline$(d)+MID$(hline$,i,2)+","
<30> 1780 WHILE LEN(dline$(d))>47
<61> 1790 dline$(d)=LEFT$(dline$(d),LEN(dline$(d))-1)
< 7> 1800 d=d+1:dline$(d)=FNnbr$(d)+"DATA "
<55> 1810 IF lnbr!+d>65534! THEN 2270
< 3> 1820 PRINT pcur$"Zeile...... "FNlnbr$(d)" wird ers
     tellt!";
<36> 1830 WEND
<11> 1840 NEXT
<10> 1850 RETURN
<45> 1860 '*------------------------------------------*
<87> 1870 '*.......... Header definieren............. *
<51> 1880 '*------------------------------------------*
<86> 1890 dline$(s+1)=FNnbr$(s+1)+"FOR adr=&H"+HEX$(adr
     (a),4)+" TO &H"+HEX$(adr(a)+byte-1,4)
< 4> 1900 dline$(s+2)=FNnbr$(s+2)+"READ a$:POKE adr,VAL
     ("+CHR$(34)+"&H"+CHR$(34)+"+a$)"
<46> 1910 dline$(s+3)=FNnbr$(s+3)+"NEXT:GOTO "+FNnbr$(d
     +1)
<44> 1920 dline$(d)=LEFT$(dline$(d),LEN(dline$(d))-1)
< 6> 1930 RETURN
<41> 1940 '*------------------------------------------*
<31> 1950 '*....... DATA-FILE SCHREIBEN........ *
<47> 1960 '*------------------------------------------*
<20> 1970 IF a THEN FOR i=1 TO a:adr(0)=MIN(adr(0),adr(
     i)):NEXT
<38> 1980 dline$(0)=FNnbr$(0)+"MEMORY &H"+HEX$(adr(0)-1
     ,4)+":RESTORE "+FNnbr$(4)
<37> 1990 a=a-1:GOSUB 1890:d=d+1
<96> 2000 MID$(dline$(s+3),INSTR(dline$(s+3),":"))=":RE
     TURN... "
<33> 2010 OPEN"O",#1,dfile$
<50> 2020 FOR i=0 TO d
< 9> 2030 PRINT #1 dline$(i)
<31> 2040 PRINT dline$(i)
<98> 2050 NEXT
< 0> 2060 CLOSE
< 1> 2070 RETURN
<36> 2080 '*------------------------------------------*
<27> 2090 '*......... Fehlerbehandlung............. *
<14> 2100 '*------------------------------------------*
<11> 2110 IF ERL<>1400 THEN 2150
<72> 2120 IF ERR=53 THEN er$="Datei nicht vorhanden!"
<33> 2130 IF ERR=64 THEN er$="Dateiname ist ungültig!"
<82> 2140 GOSUB 2200:RESUME 1350
<32> 2150 IF ERL=1460 THEN er$="Zeilennummer ist nicht
     zulässig!":GOSUB 2200:RESUME 1430
<49> 2160 IF ERL=1530 OR ERL=1560 THEN er$="Datei kann
     nicht umgewandelt werden (I/O-Fehler oder falsches
      Format).":GOTO 2180
<83> 2170 IF ERL>2010 AND ERL<2070 THEN er$="DAT-Datei
     kann nicht erzeugt werden (Disk-I/O-Fehler)."ELSE
     er$="Fehler Nr."+STR$(ERR)+" in Zeile"+STR$(ERL)
<26> 2180 er$=er$+" Programm wird abgebrochen!"
< 0> 2190 CLOSE:GOSUB 2200:RESUME 2320
<57> 2200 PRINT FNat$(0,30)CHR$(7)er$ceol$;
<24> 2210 FOR i=1 TO 4000:NEXT
<12> 2220 PRINT FNat$(0,30)SPC(89);
<92> 2230 RETURN
<28> 2240 '*------------------------------------------*
<79> 2250 '*..... Fehler: Zeilennummer zu gross..... *
<34> 2260 '*------------------------------------------*
<83> 2270 PRINT FNat$(0,30)"Fehler, Zeilennummern-Berei
     ch überschritten, "iv$">A<"nm$"bbrechen oder "iv$"
     >N<"nm$"eu beginnen?. ";
<89> 2280 a$=UPPER$(INKEY$)
<61> 2290 IF a$<>"A"AND a$<>"N"THEN 2280
<80> 2300 PRINT FNat$(0,30)SPC(90);
<59> 2310 IF a$="N" THEN CLOSE:GOTO 1090
<25> 2320 PRINT cl$:END
```

Listing HEXDAT

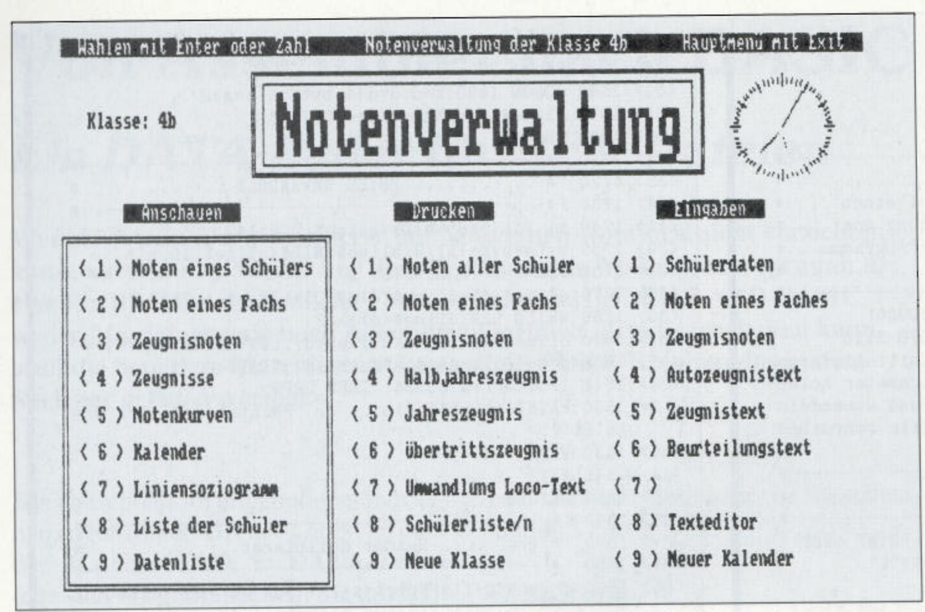


Abb1: Die Analoguhr kann ohne Probleme in eigene Programme eingebunden werden

Zeitgeschichte

Eine Analoguhr in Basic

Wenn Sie im Besitz der BASIC-Erweiterung sind, welche bei dem Gerdes-Mouse-Pack mitgeliefert wird, und Interesse daran haben, Ihre Programme etwas anspruchsvoller zu gestalten, dann sollte dieser Artikel Beachtung bei Ihnen finden.

Das Programm 'Uhr' ist eine Routine, welche es Ihnen ermöglicht, in eigenen Programmen eine Analoguhr darzustellen. Hierbei ist die Position sowie die Größe der Uhr auf dem Bildschirm frei wählbar. Sie müssen lediglich die Werte in den Zeilen 10030 und 10040 verändern. Da das Programm natürlich auch die aktuelle Uhrzeit benötigt, muß vor dem Start Ihres BASIC-Programms mittels "DATE.COM" die Uhr mit "DATE SET" gestellt werden.

Eigene Programme

Der Einbau in eigene Programme ist sichtlich einfach. Sie müssen die Uhr lediglich mit dem Befehl MERGE-"UHR.BAS" an ein eigenes Programm anhängen und in Ihrem Programm jeweils den Befehl GOSUB 10000 zum Zeichnen des Ziffernblatts und GOSUB 20000 zum Zeichnen der Uhrzeiger einbauen. Der Befehl GOSUB 20000 sollte an Stellen untergebracht werden, welche sehr oft und in gleichmäßigen Zeitabständen durchlaufen werden, da sonst die Zeiger zu ruckhaft wandern würden. Da dies bei einem laufenden Programm recht schwierig ist, empfiehlt es sich, die Uhr nur in Menüs, also dort, wo Tastaturabfragen sind, einzubauen.

Variablen

Folgende Variablen werden in dem Programm 'Uhr' genutzt:
x, y – Mittelpunkt der Uhr
b – Größe der Uhr
a, b, c, d, e,m, mu, s, ss, z, zz, m, mm
Stunde$, Minute$, Sekunde$

(Wolfgang Klein/rs)

Dieses Programm ist in Mallard-BASIC geschrieben. Nach dem Eingeben sollte es zuerst mit SAVE "UHR.BAS" <RETURN> abgespeichert werden. Wollen Sie die Routine in eigene Programme einbinden, müssen Sie darauf achten, daß die Zeilen 10-40 nicht in dem Programm UHR.BAS vorhanden sind, da sonst Ihr Programm überschrieben werden könnte.

```
LISTING >UHR      <, REMARK = >'<.

<52> 10 PRINT CHR$(27)"f":CLS:' Uhr.bas.. von Wolfgang
     Klein
<77> 20 GOSUB 10000
<88> 30 GOSUB 20000
< 7> 40 GOTO 30
<64> 10000 '*******
< 9> 10010 '+... Ziffernblatt.... +
<72> 10020 '*******
<96> 10030 x=400:y=300:' #######.... Mittelpunkt der Uh
     r z.B. x=640:y=430 für rechts oben
<43> 10040 b=140:' #######..... Größe der Uhr. z.B. 70
     für rechts oben
<77> 10050 a=3.141592/6
<50> 10060 PLOT x,y,1
<96> 10070 c=b/14:d=6*c:e=c*10:f=12*c
<30> 10080 FOR n=0 TO 6.3 STEP a/5
<62> 10090 GOSUB 10140
<82> 10100 NEXT
<51> 10110 c=3*c:b=b/140*130
<50> 10120 FOR n=0 TO 6 STEP a
<20> 10130 GOSUB 10140:NEXT
< 2> 10140 MOVE x,y
<67> 10150 MOVER COS(n)*b,SIN(n)*b
<72> 10160 DRAWR COS(n)*c,SIN(n)*c
<69> 10170 RETURN
< 6> 20000 GOSUB 20300
<69> 20010 '*******
< 7> 20020 ' +....... Stundenzeiger...... +
```

Listing UHR

```
<77> 20030 '*******
< 0> 20040 MOVE x,y
<55> 20050 DRAWR SIN(s)*d,COS(s)*d,1
< 8> 20060 MOVE x,y
<69> 20070 IF s=ss THEN GOTO 20130
<52> 20080 DRAWR SIN(ss)*d,COS(ss)*d,0
<18> 20090 ss=s
<68> 20100 '*******
<68> 20110 '+........ Minutenzeiger...... +
<76> 20120 '*******
<98> 20130 MOVE x,y
<43> 20140 DRAWR SIN(m)*e,COS(m)*e,1
< 7> 20150 MOVE x,y
<55> 20160 IF m=mm THEN GOTO 20220
<58> 20170 DRAWR SIN(mm)*e,COS(mm)*e,0
<65> 20180 mm=m
< 5> 20190 '*******
<29> 20200 '+....... Sekundenzeiger...... +
<75> 20210 '*******
<97> 20220 MOVE x,y
<56> 20230 DRAWR SIN(z)*f,COS(z)*f,1
< 6> 20240 MOVE x,y
< 3> 20250 IF z=zz THEN GOTO 20000
<91> 20260 DRAWR SIN(zz)*f,COS(zz)*f,0
<92> 20270 zz=z
<77> 20280 RETURN
< 8> 20290 '*******
<76> 20300 '+....... Zeit holen......... +
<78> 20310 '*******
< 5> 20320 Stunde$=HEX$(PEEK(-1034),2):Minute$=HEX$(PEE
     K(-1033),2):Sekunde$=HEX$(PEEK(-1032),2)
<81> 20330 z= VAL(sekunde$):z=(a/5)*z
<44> 20340 mu= VAL(minute$):m=(a/5)*mu
<82> 20350 s= VAL(stunde$):s=a*(s+mu/60)
<72> 20360 RETURN
```

Listing Uhr

Superpack-Aktion · Superpack-Aktion

Ja, ich bestelle aus Ihrer Zeitschriften-Aktion folgende Produkte

☐ **Best.-Nr. 40** **CPC-Superpack** 4 Hefte **28,–** DM

☐ **Best.-Nr. 35** **PC-Amstrad-Ultrapack**
12 × PC-Amstrad/Schneider der Ausgaben 1/87 – 12/87
3 Ausgaben PC International (7/86 – 12/86)
2 praktische Sammelordner für nur **50,–** DM
(nur solange Vorrat reicht)

Demodisketten:

☐ Best.-Nr. 005	ConText PRO	5,– DM	☐ Best.-Nr. 003	ERGO 3.0	5,– DM
☐ Best.-Nr. 006	TOOLBOX-Spezial	5,– DM	☐ Best.-Nr. 001	Fraktal-3D + MaskEdit	5,– DM
☐ Best.-Nr. 007	3D-Draw	5,– DM			

☐ Einen Verrechnungsscheck in Höhe des Rechnungsbetrages habe ich beigefügt.

☐ Ich bitte um Lieferung per Nachnahme, zuzügl. der Nachnahmegebühr (nur innerhalb der BRD)

Gesamtbetrag ________ DM

Datum ________ Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

PC-Bestellservice

Hiermit mache ich von Ihrem Angebot Gebrauch und bestelle:

SOFTWARE 5 1/4"		SOFTWARE 3 1/2"			DM
2282	☐	2281	☐	**ConText EASY,** (MS-DOS)	99,–
2407	☐	2408	☐	**ConText PRO,** (MS-DOS)	199,–
2409	☐	2410	☐	**Upgrade - ConText PRO**	129,–
2415	☐	2416	☐	**ConText-ADREVA,** Adreßverwaltung	49,–
2650	☐	2651	☐	**DMV-Power-Pack**	349,–
2615	☐	2616	☐	**H.A.I.-Tech,** (MS-DOS)	199,–
2522	☐	2521	☐	**ERGO 3.0,** DOS-Benutzeroberfläche	199,–
2524	☐	2523	☐	**Update ERGO,** Version 3.0	69,–
229	☐	235	☐	**Hyperkey,** Version 4.0, (MS-DOS)	99,–
256	☐	257	☐	**VirusDoktor,** (MS-DOS)	99,–
238	☐	239	☐	**MaskEdit Plus I,** (Turbo Pascal 3/4/5)	99,–
2611	☐	2612	☐	**MaskEdit Plus II,** (Turbo C, Microsoft C)	99,–
2619	☐	2620	☐	**MaskEdit Plus III,** (TopSpeed Modula-2)	99,–
2613	☐	2614	☐	**MaskEdit Plus,** Paket I + II	149,–
2621	☐	2622	☐	**MaskEdit Plus,** Paket I bis III	199,–
227	☐	233	☐	**BCI DisAsm/86,** (MS-DOS)	199,–
2402	☐	2401	☐	**TopSpeed Modula-2,** Compiler	348,–
2404	☐	2403	☐	**TopSpeed Modula-2,** TechKit	248,–
2406	☐	2405	☐	**TopSpeed Modula-2,** Debugger	248,–
2412	☐	2411	☐	**Paket Compiler + Debugger**	555,–
2413	☐	2412	☐	**TopSpeed Modula-2,** OS/2-Version	698,–
6001	☐	6002	☐	**DOS-Tools 1,** RSM-Manager	69,–
6003	☐	6004	☐	**DOS-Tools 2,** PC-Utilities	69,–
6005	☐	6006	☐	**DOS-Tools 3,** MS-DOS-Utilities	49,–
6007	☐	6008	☐	**DOS-Tools 4,** 3D-Draw	99,–
6009	☐	6010	☐	**DOS-Tools 5,** GW-Basic-Toolbox-Oberfläche	69,–
6011	☐	6012	☐	**DOS-Tools 6,** GW-Basic-Toolbox-DOS-Funktionen	69,–
6013	☐	6014	☐	**DOS-Tools 7,** GW-Basic-Toolbox-Hercules-Grafik	69,–
6015	☐	6016	☐	**DOS-Tools 8,** NAVIGATOR für FS III	69,–

SOFTWARE 5 1/4"		SOFTWARE 3 1/2"			DM
20500	☐	20501	☐	**PCpur Software I,** MultiFont Plus	69,–
20502	☐	20503	☐	**PCpur Software Ia,** Zusatzfonts MultiFont Plus	39,–
20504	☐	20505	☐	**PCpur Software II,** ASCII-Tiger, Programmeditor	69,–
2801	☐	2802	☐	**toolbox Spezial I,** Turbo Prolog	35,–
2804	☐	2805	☐	**toolbox Spezial II,** PASCOMP	35,–
2806	☐	2807	☐	**toolbox Spezial III,** Echtzeit	35,–
2808	☐	2809	☐	**toolbox Spezial IV,** CALC	35,–
2811	☐	2812	☐	**toolbox Spezial V,** STRUKTO	35,–
2813	☐	2814	☐	**toolbox Spezial VI,** TP4/5-Spiele	35,–
2815	☐	2816	☐	**toolbox Spezial VII,** dt. Silbentrennung	35,–
2817	☐	2818	☐	**toolbox Spezial VIII,** Spieltheorie	35,–
2819	☐	2820	☐	**toolbox Spezial IX,** HPGL-Plotter	35,–
2821	☐	2822	☐	**toolbox Spezial X,** CRC-Sicherung	35,–
2823	☐	2824	☐	**toolbox Spezial XI,** MAP, MARK & RELEASE	35,–
2827	☐	2828	☐	**toolbox Spezial XIII,** Trickkiste Vol. I	35,–
2829	☐	2830	☐	**toolbox Spezial XIV,** Trickkiste Vol. 2	35,–
2831	☐	2832	☐	**toolbox Spezial XV,** Fitting & Lin.-Glei.	35,–
2833	☐	2834	☐	**toolbox Spezial XVI,** nat.-sprachl. Schnittst.	35,–
236	☐	237	☐	**MAUSALL,** (MS-DOS)	29,90
246	☐	247	☐	**C-Funktionssammlung**	99,–
2952	☐	———		**GEMInterface PC,** Turbo Pascal 4 + 5.0	198,–
2954	☐	———		**GEMInterface PC,** TopSpeed Modula-2	198,–
129	☐	185	☐	**PC-Spielebox No.1**	49,–
1291	☐	1292	☐	**PC-Spielebox No.2**	49,–
161	☐	162	☐	**KNOW-PC, Unterhaltungsspiel**	49,–
1293	☐	1294	☐	**Ergänzungsfragen zu KNOW-PC**	29,–
248	☐	249	☐	**Fraktal-Generator 3D,** MS-DOS	69,–

+ Porto/Verpackung
(Inland 4,– DM, Ausland 6,– DM) ________ = Gesamtbetrag ________

»Kleinanzeigen-Markt«

Private Anzeigen: Nur DM 5,– je angefangene Zeile.
Geschäftliche Empfehlungen: DM 8,– je angefangene Zeile, zzgl. gesetzlicher Mehrwertsteuer.
Kreuzen Sie bitte an, in welche Rubrik (s. Karte) Ihre Anzeige gehört, schreiben Sie Ihren Text in die Karte (jedes Kästchen = ein Zeichen; Buchstabe, Satzzeichen oder Wortzwischenraum).
Achtung! Der Abdruck erfolgt **nur** gegen Vorkasse (Verrechnungsscheck).
Bitte veröffentlichen Sie meine Anzeige in der nächsterreichbaren **»PC International«** für
☐ private Zwecke ☐ gewerbliche Zwecke (gewerbliche Anzeigen werden mit **G** gekennzeichnet)

Das ist der Text: *(Bitte deutlich in Druckbuchstaben schreiben!)*

☐ Die Anzeige soll als Chiffre-Anzeige erscheinen (nur möglich bei Privat-Anzeige)
Chiffre-Gebühr 10,– DM zzgl. zum Anzeigenpreis

In dieser Rubrik:

☐ **Biete an**	☐ **Suche**	☐ **Tausch**	☐ **Stellenmarkt/freie Mitarbeit**
☐ Hardware	☐ Hardware		☐ **Geschäftsverbindungen**
☐ Software	☐ Software		☐ **Verschiedenes**

Bei Angeboten:
Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze.

Datum ________ Unterschrift (bei Minderjährigen des gesetzl. Vertreters)

»JOYCE-Bestellservice« Knüller für JOYCE-Fans

Ich mache von Ihrem Angebot Gebrauch und bestelle hiermit:

30170	____ Stck. **Joyce-Sonderheft-Superpack**	3 x Joyce-Sonderhefte der Ausgaben 2/87 – 4/88	**29,– DM**
30171	____ Stck. **Joyce-Sonderheft-Superpack-Databox**	5 x 3 1/2"-Disketten zu den Ausgaben 2/87 – 4/88	**79,– DM**
215	____ Stck. **JOYCE Programmsammlung Vol. 1**		**49,– DM**
216	____ Stck. **JOYCE Programmsammlung Vol. 2**		**49,– DM**
217	____ Stck. **JOYCE Programmsammlung Vol. 3**		**59,– DM**
219	____ Stck. **JOYCE Programmsammlung Vol. 4**		**49,– DM**
220	____ Stck. **JOYCE Programmsammlung Vol. 5**		**99,– DM**
221	____ Stck. **JOYCE Programmsammlung Vol. 6**		**59,– DM**
222	____ Stck. **JOYCE Programmsammlung Vol. 7**		**69,– DM**

+ Porto/Verpackung
(Inland 4,– DM, Ausland 6,– DM) ________

Gesamtbetrag ________

☐ Den Betrag bezahle ich mittels des beigefügten Verrechnungsschecks.
☐ Ich bitte um Lieferung per Nachnahme zuzüglich der Nachnahmegebühr. (Bei Lieferung ins Ausland ist Nachnahme nicht möglich.)

Datum ________ Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)



»Kleinanzeigen-Markt«

Absender: *(Bitte genaue Anschrift angeben!)*

Name

Vorname

Firma

Straße/Nr./Postfach

PLZ/Ort

Bitte ausreichend frankieren

Antwortkarte

DMV-Verlag
PC International
Postfach 250

3440 Eschwege



»Superpack«

Absender: *(Bitte genaue Anschrift angeben!)*

Name

Vorname

Firma

Straße/Nr./Postfach

PLZ/Ort

Bitte ausreichend frankieren

Antwortkarte

DMV-Verlag
»Superpack«
Postfach 250

3440 Eschwege



»JOYCE-Bestellservice«

Absender: *(Bitte genaue Anschrift angeben!)*

Name

Vorname

Firma

Straße/Nr./Postfach

PLZ/Ort

Bitte ausreichend frankieren

Antwortkarte

DMV-Verlag
PC International
Postfach 250

3440 Eschwege



»PC-Bestellservice«

☐ Ich mache von Ihrem Angebot Gebrauch und bestelle hiermit die umseitig ausgewählten Produkte.
Bitte liefern Sie die Produkte an die folgende Anschrift:

Name

Straße/Postfach

PLZ/ORT

☐ Den Betrag bezahle ich mittels des beigefügten Verrechnungsschecks.

☐ Ich bitte um Lieferung per Nachnahme zuzüglich der Nachnahmegebühr (nur innerhalb der BRD).

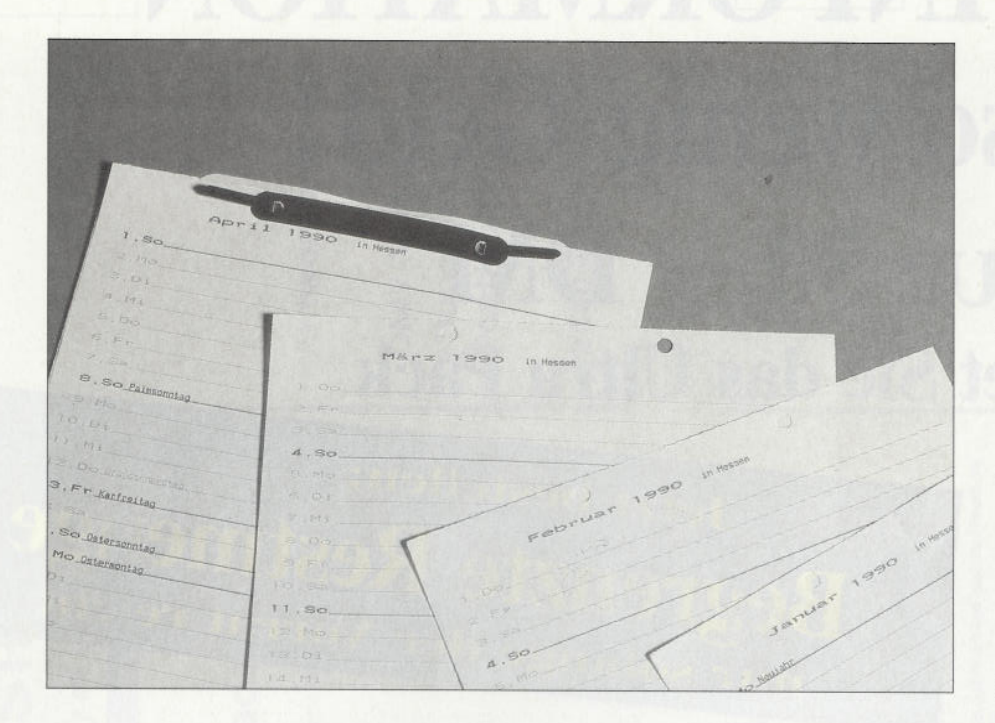
Datum — Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

Bitte ausreichend frankieren

Antwortkarte

DMV-Verlag
PC International
Postfach 250

3440 Eschwege



Alle Termine im Blick

Ein Kalender mit vielen Extras

Alle Jahre wieder kommt die gleiche Prozedur auf uns zu. Es geht darum, Geburts- und sonstige Feiertage in einen zuvor teuer gekauften Kalender zu übertragen, um sie dann später doch zu vergessen. Dies hat spätestens jetzt ein Ende. Mit diesem Kalender-Programm ist es möglich, sich seine eigenen, ganz besonderen Kalender zu drucken.

Dabei wird nicht – so wie bei vielen anderen Programmen – nur der Wochentag und das Datum berechnet, sondern es werden auch die Feiertage für das gewünschte Jahr ermittelt.

Aufgrund einer unterschiedlichen Feiertagsregelung in den Bundesländern wurde das Programm so gestaltet, daß es für jedes Bundesland die Feiertage 'kennt'. Dazu muß lediglich das entsprechende Land in einem Menü ausgewählt werden.

Natürlich wurde auch die Umstellung der Uhren auf Sommer- und Winterzeit und einige Gedenktage wie Muttertag, Totensonntag und so weiter berücksichtigt.

Zusätzlich können auch noch persönliche Daten, zum Beispiel Geburtstage, Hochzeitstag und dergleichen in den Kalender eingefügt werden. Hierzu müssen diese Daten dem Programm als DATA-Zeilen angehängt werden.

Als Beispiel hierfür wurden einige Namenstage als persönliche Daten hinzugefügt.

Programmablauf

Nach dem Starten des Programms wird auf der linken Bildschirmseite eine Menüspalte aufgebaut, und das Programm wartet auf die Eingabe einer Jahreszahl für den zu berechnenden Kalender. Ist man dem nachgekommen, gelangt man in das Menü zur Auswahl des Bundeslandes. Nach der Wahl des Landes dauert es einige Sekunden, bis der Kalender auf dem Monitor erscheint, da hier einige Berechnungen angestellt werden.

Bei der Bildschirmausgabe wird immer nur ein Monat dargestellt, wobei die Daten der einzelnen Tage untereinanderstehen. Auf der linken Seite stehen das Datum, Wochentag und die Namen für Feier- und Gedenktage. Die persönlichen Daten werden am Zeilenende eingefügt. Sonntage und gesetzlich Feiertage werden bei der Ausgabe invertiert. In der ersten Zeile erscheinen zusätzlich noch die Angaben über das Land, Monat, Jahr und ob es sich um ein Schaltjahr handelt.

Nach dem Aufbau des Bildschirms steht folgendes zur Auswahl:

1. Mit den Tasten 1 bis ↑ (1=Januar bis ↑=Dezember) kann jeder gewünschte Monat direkt angewählt werden.
2. Durch Druck auf die [+]- oder [-]-Taste wird das nächste bzw. das vergangene Jahr berechnet.
3. D = Kalender drucken, hiermit kommt man in ein Drucker-Menü, in welchem zwei verschiedene Möglichkeiten des Ausdrucks bestehen.

3.1 Jeder Monat erhält sein eigenes Kalenderblatt, wie bei der Bildschirmausgabe. Hierbei werden Sonntage und Feiertage durch Doppel- und Fettdruck hervorgehoben. Der Ausdruck ist nur auf 12-Zoll- Endlos-Papier möglich.

3.2 Hierbei wird das gesamte Jahr auf einem Blatt ausgedruckt und die gesetzlichen Feiertage am Seitenende aufgelistet.

4. N = Eingabe einer neuen Jahreszahl.
5. EXIT = Programm beenden.
6. (e) - (k) soll daran erinnern, das Feiertage mit einem vorangestellten Buchstaben nur in Gemeinden mit überwiegend evangelischer oder katholischer Bevölkerung gültig sind.

Eingabe eigener Daten

Hinzuzufügen wäre eigentlich nur noch, daß sämtliche Feier- und Gedenktage in DATA-Zeilen untergebracht wurden und somit nach Belieben erweitert oder entfernt werden können. Nur die Endkennzeichnung jedes einzelnen Datenblocks darf nicht entfernt werden, da es sonst zu einer Fehlermeldung des Computers kommt.

Die Kennzeichnug der gesetzlichen Feiertage (ab Zeile 3460 beziehungsweise 3820) in den Bundesländern erfolgt mit Hilfe eines elf Zeichen langen Strings, wobei für jedes Land ein Zeichen steht.

Hierbei bedeutet :

0 = kein Feiertag
x = gesetzlicher Feiertag
e = in Gemeinden mit überwiegend evangelischer Bevölkerung
k = in Gemeinden mit überwiegend katholischer Bevölkerung

Bei den Gedenktagen ist es ausreichend, wenn an dieser Stelle nur ein Leerzeichen eingefügt wird.

Bei der Druckerausgabe wurde das Ändern der Schriftart, zum Beispiel Breitschrift, Doppeldruck und so weiter,

nicht mit den entsprechenden Steuercodes getan, sondern mit der Möglichkeit des JOYCE-Druckers, eine gemischte Schriftart zu wählen.

Nähere Informationen hierüber stehen im Handbuch 1. Anhang 2.

Wer sich näher damit befassen möchte, sollte doch einmal folgendes kleine Programm ausprobieren.

```
10 WIDTH LPRINT 255
20 OPTION NOT TAB ' ist wichtig bei der Übergabe von CHR$(9), da sonst ein Tabulatorsprung ausgeführt wird.
30 FOR i=0 TO 63
40 LPRINT CHR$(27)+"!"+CHR$(i); "Schriftarten"
50 NEXT
60 LPRINT CHR$(27)+"!"+CHR$(0)
```

Der Ausdruck auf 12-Zoll-Endlospapier wurde deshalb gewählt, damit auf den Zeilen hinter den Tagesangaben auch noch handschriftliche Eintragungen vorgenommen werden können. Bei einem einzeiligen Zeilenabstand ist der Ausdruck auch auf Einzelblatt möglich.

Alle Zeilen, die mit einem REM-Zeichen beginnen, werden vom Programm nicht angesprungen, müssen daher auch nicht eingegeben werden.

(Herbert Tandler/rs)

Dieses Programm ist in Mallard-Basic geschrieben.

Nach dem Eingeben sollte es erst mit SAVE"KALENDER.BAS" <RETURN> abgespeichert werden. Eventuelle Veränderungen können nun vorgenommen werden, und das Programm sollte wieder abgespeichert werden.

Später kann es dann unter BASIC mit RUN"KALENDER.BAS" <RETURN> und unter CP/M mit A>BASIC KALENDER <RETURN> eingeladen werden.

```
LISTING >KALENDER<, REMARK = >'<.

< 1> 10 '
<67> 20 '. K A L E N D E R. von. 1600. bis. 2199
< 3> 30 '
< 4> 40 ' written 1989 by H.Tandler
< 5> 50 '
<51> 60 '----------------------------------------
< 7> 70 '
< 8> 80 ' Initialisierung. - Bildschirm
< 9> 90 '
<16> 100 esc$=CHR$(27) :' escape
<94> 110 bell$=CHR$(7) :' piepston
<64> 120 cls$=esc$+"E"+esc$+"H" :' Bildschirm löschen
<20> 130 con$=esc$+"e":coff$=esc$+"f":' Cursor ein-aus
<87> 140 invon$=esc$+"p":invoff$=esc$+"q":' Invers ein-
     aus
<94> 150 staton$=esc$+"1":statoff$=esc$+"0":' Statuszei
     le ein-aus
<72> 160 bildumon$=esc$+"v":bildumoff$=esc$+"w":'Bildum
     lauf am Zeilenende ein-aus
<42> 170 DEF FNloc$(z,s)=esc$+"Y"+CHR$(31+z)+CHR$(31+s)
     :' Cursor auf Position (zeile,spalte)
<13> 180 DEF FNwin$(oz,ls,uz,rs)=esc$+"X"+CHR$(31+oz)+C
     HR$(31+ls)+CHR$(31+uz)+CHR$(31+rs) :' Window defin
     ieren
<82> 190 WIDTH 255,90:PRINT bildumoff$;statoff$;cls$
<19> 200 '
<21> 210 ' Initialisierung. - Drucker - Einstellung auf
      endlos Papier
<23> 220 '
<88> 230 DEF FNgem$(n)=esc$+"!"+CHR$(n). ' wahl einer g
     emischten Schriftart (n =0-63)
< 7> 240 DEF FNzese$(n)=esc$+"C"+CHR$(n). ' Zeilen pro
     Seite
<34> 250 DEF FNleze$(n)=esc$+"N"+CHR$(n). ' Leere Zeile
     n am Seitenende
< 4> 260 zei.6$=esc$+"2". ' 6 Zeilen pro Zoll
<87> 270 zei.8$=esc$+"0". ' 8 Zeilen pro Zoll
<75> 280 endlos$=esc$+"c". ' Endlospapier
<52> 290 einblatt$=esc$+"$". ' Einzelblatt
<17> 300 pica$=CHR$(18). ' Schrift PICA
<97> 310 un.on$=esc$+"-1": un.off$=esc$+"-0". ' Unterst
     reichung ein - aus
<51> 320 pendon$=esc$+"9": pendoff$=esc$+"8". ' Papiere
     nde anzeigen ein - aus
<24> 330 entwurf$=esc$+"x0". ' Entwurfs-Qualität
<10> 340 default$=esc$+"d". ' Neubestimmung der Drucker
     -Standardeinstellung
<42> 350 ptr.reset$=esc$+"§". ' Drucker auf Standardein
     stellung zurücksetzen
<63> 360 WIDTH LPRINT 255
<54> 370 LPRINT pica$+FNzese$(72)+FNleze$(1)+zei.6$+end
     los$+pendon$+default$+ptr.reset$+CHR$(13);
<36> 380 '
<38> 390 ' Dimensionieren der Felder
<21> 400 '
< 9> 410 DIM mon$(12),mon(12),land$(11)
<25> 420 '
<27> 430 ' Daten für Wochentage, Monate und Länder einl
     esen
<29> 440 '
<73> 450 RESTORE 3240
<25> 460 FOR i=0 TO 6: READ wt$(i):NEXT
```

Listing KALENDER

```
<27> 470 FOR i=1 TO 12:READ mon$(i),mon(i):NEXT
<57> 480 FOR i=1 TO 11:READ land$(i):NEXT
<39> 490 '
<22> 500 ' Titelbild - Menue
<24> 510 '
<91> 520 PRINT cls$
<92> 530 FOR i=1 TO 32
<97> 540 PRINT FNloc$(i,1);invon$;SPC(9);CHR$(149);invo
     ff$;
<48> 550 NEXT
< 2> 560 a$=STRING$(9,154)+CHR$(157)
<24> 570 PRINT FNloc$(2,2);invon$;"KALENDER"
<27> 580 PRINT:PRINT " 1989 by "
<11> 590 PRINT "H.Tandler"
<74> 600 PRINT a$
<74> 610 PRINT " Monat ="
< 9> 620 PRINT " Tasten"
<57> 630 PRINT " 1 bis ^"
<82> 640 PRINT a$
<96> 650 PRINT "Ä+Ü.. Ä-Ü"
<75> 660 PRINT " + od. -"
<32> 670 PRINT " 1- Jahr"
<90> 680 PRINT a$
<83> 690 PRINT ".. D ="
<22> 700 PRINT " Kalender"
<98> 710 PRINT " drucken"
<79> 720 PRINT a$
<24> 730 PRINT ".. N ="
<85> 740 PRINT " Neues"
<82> 750 PRINT " Jahr"
<87> 760 PRINT a$
<66> 770 PRINT " EXIT ="
<88> 780 PRINT " Programm"
<20> 790 PRINT " beenden"
<76> 800 PRINT a$
<38> 810 PRINT " (e)-(k)"
<85> 820 PRINT "bei über-"
<58> 830 PRINT "wiegender"
<34> 840 PRINT " evang.-"
<48> 850 PRINT " katho."
<56> 860 PRINT "Bevölkerg";invoff$;
<26> 870 PRINT FNwin$(1,11,32,90);cls$
<41> 880 '
<43> 890 ' Eingabe der Jahreszahl
<26> 900 '
<56> 910 PRINT FNloc$(15,10);:INPUT"Bitte das gewünscht
     e Jahr eingeben ( 1600 bis 2199 ) : ",jahr$
<53> 920 jahr=VAL(jahr$)
<68> 930 IF jahr<1600 OR jahr>2199 THEN PRINT:PRINT TAB
     (5);bell$;invon$;" Es wurde ein nicht zulässiges J
     ahr eingegeben ";invoff$:GOTO 910
<34> 940 '
<36> 950 ' Bundesland wählen
<38> 960 '
< 6> 970 PRINT cls$
<57> 980 FOR i=1 TO 11
<49> 990 PRINT FNloc$(3+i*2,30);DEC$(i,"##");" = ";land
     $(i)
<82> 1000 NEXT
<31> 1010 PRINT FNloc$(29,20);:INPUT "Bitte die Nummer
     des gewünschten Landes eingeben ";land$
<45> 1020 land=VAL(land$)
<26> 1030 IF land<1 OR land>11 THEN PRINT FNloc$(31,20)
     ;bell$;invon$;" Fehler bei der Eingabe (1-11) ";in
```

Listing KALENDER

```
        voff$:GOTO 1010
<43> 1040 mo=1
<27> 1050 DIM tag$(12,31),tag(12,31),dat$(12,31):mon(2)
     =28
<56> 1060 RESTORE 3460
<26> 1070 PRINT cls$
<65> 1080 PRINT FNloc$(15,15);"Bitte etwas Geduld, die
     Feiertage werden berechnet.";
<11> 1090 '
<85> 1100 ' auf Schaltjahr prüfen
<88> 1110 '
<71> 1120 IF jahr MOD 4 = 0 OR jahr MOD 400 = 0 THEN mo
     n(2)=29
<62> 1130 IF jahr MOD 100 = 0 AND jahr MOD 400 <>0 THEN
      mon(2)=28
<97> 1140 '
< 1> 1150 ' Datum für das Osterfest berechnen. nach C.F
      Gauß
< 4> 1160 '
<27> 1170 M=22 : N=2
<44> 1180 IF jahr >=1700 THEN M=23 : N=3
< 0> 1190 IF jahr >=1800 THEN M=23 : N=4
<56> 1200 IF jahr >=1900 THEN M=24 : N=5
<73> 1210 IF jahr >=2100 THEN M=24 : N=6
<55> 1220 a= jahr MOD 19
<25> 1230 b= jahr MOD 4
<88> 1240 c= jahr MOD 7
<17> 1250 d= (19*a+M) MOD 30
<41> 1260 e= (2*b+4*c+6*d+N) MOD 7
<97> 1270 osta= 22+d+e: IF osta<=31 THEN osmo=3 :GOTO 1
     340
<74> 1280 osta= d+e-9 :osmo=4
< 4> 1290 IF osta > 25 THEN osta=osta-7
<79> 1300 IF osta=25 AND d=28 AND a>10 THEN osta=osta-7
<92> 1310 '
<95> 1320 ' Feste und Feiertage, die sich auf das Oster
     fest beziehen
<98> 1330 '
<32> 1340 mon=osmo
<89> 1350 READ tag,fe$,na$
<76> 1360 IF fe$= "ENDE" THEN 1450
<31> 1370 tag=osta+tag
<38> 1380 GOSUB 1850
<91> 1390 GOSUB 2010
<25> 1400 tag$(mon,tag)=" "+na$+" "
<91> 1410 GOTO 1340
<97> 1420 '
< 1> 1430 ' bewegliche Feiertage, denen das Kirchenjahr
      zu Grunde liegt
< 4> 1440 '
<14> 1450 tag=25:mon=12:GOSUB 1910:IF wt=0 THEN wt=7
<81> 1460 mon=11:tag=25-wt-36:GOSUB 1850
<29> 1470 vota=tag
< 8> 1480 mon=11
< 4> 1490 READ tag,fe$,na$
<47> 1500 IF fe$="ENDE" THEN 1590
<43> 1510 tag=vota+tag
<24> 1520 GOSUB 1850
<77> 1530 GOSUB 2010
<39> 1540 tag$(mon,tag)=" "+na$+" "
<70> 1550 GOTO 1480
<12> 1560 '
<15> 1570 ' sonstige Feiertage
<18> 1580 '
< 9> 1590 READ mon,tag,fe$,na$
<21> 1600 IF fe$="ENDE" THEN 1670
<73> 1610 GOSUB 2010
<45> 1620 tag$(mon,tag)=tag$(mon,tag)+" "+na$+" "
<91> 1630 GOTO 1590
< 8> 1640 '
<11> 1650 ' sonstige bewegliche Feiertage. - Sommer- W
     interzeit
<14> 1660 '
<83> 1670 READ mon,tag,na$
<49> 1680 IF na$="ENDE" THEN 1750
< 4> 1690 GOSUB 1910
<43> 1700 tag$(mon,tag-wt)=tag$(mon,tag-wt)+" "+na$+" "
<73> 1710 GOTO 1670
< 4> 1720 '
< 7> 1730 ' einlesen von persönlichen Daten
<10> 1740 '
<79> 1750 READ mon,tag,na$
<76> 1760 IF na$="ENDE" THEN 2100
<12> 1770 dat$(mon,tag)=dat$(mon,tag)+" "+na$
<80> 1780 GOTO 1750
<25> 1790 '
< 0> 1800 ' Unter Programme
< 3> 1810 '
< 6> 1820 '
< 9> 1830 ' Datum prüfen
<12> 1840 '
<45> 1850 IF tag>mon(mon) THEN tag=tag-mon(mon):mon=mon
     +1:GOTO 1850
<47> 1860 IF tag<1 THEN tag=tag+mon(mon-1):mon=mon-1:GO
     TO 1860
<16> 1870 RETURN
<24> 1880 '
<27> 1890 ' Wochentag ermitteln
< 2> 1900 '
```

Listing KALENDER

```
<93> 1910 d=0
< 0> 1920 FOR i=0 TO mon-1
<10> 1930 d=d+mon(i)
<13> 1940 NEXT
<57> 1950 d=d+tag:ja=jahr-1
<78> 1960 wt=INT(ja+INT(ja/4)-INT(ja/100)+INT(ja/400)+d
     ) MOD 7
<18> 1970 RETURN
<26> 1980 '
<29> 1990 ' Fusszeile für Feiertage in ev. oder kath. G
     emeinden
<84> 2000 '
<88> 2010 IF MID$(fe$,land,1)="0" OR MID$(fe$,land,1)="
     "THEN RETURN
<95> 2020 IF MID$(fe$,land,1)="e" OR MID$(fe$,land,1)="
     k"THEN tag(mon,tag)=1:na$="("+MID$(fe$,land,1)+")
     "+na$:GOTO 2050
<53> 2030 IF MID$(fe$,land,1)="x" THEN tag(mon,tag)=1
<91> 2040 RETURN
<40> 2050 fuze$=" (e) = Feiertag in Gemeinden mit überw
     iegend evangelischer Bevölkerung"+CHR$(13)+CHR$(10
     )+" (k) = Feiertag in Gemeinden mit überwiegend ka
     tholischer Bevölkerung"
<13> 2060 tag(mon,0)=1:RETURN
< 6> 2070 '
< 9> 2080 ' Kalender auf Bildschirm ausgeben
<12> 2090 '
<57> 2100 mon=mo:menu$="1234567890β^ND"+CHR$(22)+CHR$(2
     8)+CHR$(27)
<37> 2110 tag=1:GOSUB 1910
<80> 2120 PRINT cls$;FNloc$(0,1);
<47> 2130 PRINT invon$;SPC(20);land$(land);". "mon$(mon
     );jahr;
<35> 2140 IF mon(2)=29 THEN PRINT ".. Schaltjahr";
<56> 2150 PRINT SPACE$(45);invoff$
<55> 2160 a=32-mon(mon)
< 0> 2170 FOR i=1 TO mon(mon)
<33> 2180 c=LEN(tag$(mon,i))+LEN(dat$(mon,i))
<85> 2190 PRINT FNloc$(a+i,1);" ";
<26> 2200 IF wt=0 OR tag(mon,i)=1 THEN PRINT invon$;
<62> 2210 PRINT DEC$(i,"##.");wt$(wt);tag$(mon,i);STRIN
     G$(74-c,".");dat$(mon,i);invoff$
<92> 2220 wt=wt+1:IF wt=7 THEN wt=0
<96> 2230 NEXT
  5> 2240 PRINT bell$;
< 4> 2250 '
< 7> 2260 ' Tastatur abfrage Menue
<10> 2270 '
<85> 2280 a$=INKEY$:IF a$="" THEN 2280
<78> 2290 a=INSTR(menu$,UPPER$(a$)):IF a=0 THEN PRINT b
     ell$;:GOTO 2280
<89> 2300 IF a>12 THEN ON a-12 GOTO 2350,2390,3090,3130
     ,3170
<30> 2310 mon=a:GOTO 2110
<96> 2320 '
< 0> 2330 ' Neues Jahr berechnen
< 3> 2340 '
<97> 2350 RUN
< 9> 2360 '
<12> 2370 ' Kalender drucken
<15> 2380 '
<87> 2390 PRINT cls$:wahl$="123"
<26> 2400 PRINT FNloc$(4,10);"Unter dem Menüpunkt 1. we
     rden alle dem Programm bekannten"
<73> 2410 PRINT TAB(10);"Feiertage und persönlichen Dat
     en ausgedruckt."
<13> 2420 PRINT TAB(10);"Hierzu muß der Blattanfang auf
      die Oberkante des Farbbandes"
<49> 2430 PRINT TAB(10);"positioniert werden."
<82> 2440 PRINT FNloc$(9,10);"Bei Punkt 2. werden nur d
     ie gesetzlichen Feiertage am"
<34> 2450 PRINT TAB(10);"Seitenende aufgelistet."
<12> 2460 PRINT FNloc$(12,10);"Der Ausdruck erfolgt in
     Entwurfsqualität, zum besseren"
<87> 2470 PRINT TAB(10);"erkennen der Feiertag."
<10> 2480 PRINT FNloc$(17,10);"1. für jeden Monat ein B
     latt (auf Endlospapier)"
<65> 2490 PRINT FNloc$(19,10);"2. Jahresübersicht auf e
     inem Blatt (Einzelblatt)"
<68> 2500 PRINT FNloc$(21,10);"3. Menue"
<86> 2510 PRINT FNloc$(24,10);"Bitte wählen : ";
<73> 2520 a$=INKEY$:IF a$="" THEN 2520
<30> 2530 wahl=INSTR(wahl$,a$):IF wahl=0 THEN PRINT bel
     l$;:GOTO 2520
<96> 2540 ON wahl GOTO 2580,2790,2110
<10> 2550 '
<13> 2560 ' Drucken (endlospapier)
<16> 2570 '
<92> 2580 tag=1:mo=mon:mon=1:GOSUB 1910
< 6> 2590 PRINT cls$;FNloc$(10,10);:INPUT"Wenn Drucker
     bereit >RETURN< drücken ",a$
< 2> 2600 FOR i=1 TO 12
<34> 2610 LPRINT:LPRINT:LPRINT
<31> 2620 LPRINT FNgem$(58);TAB(10);mon$(i);jahr;FNgem$
     (25);" in ";land$(land)
<64> 2630 LPRINT:LPRINT
<21> 2640 FOR j=1 TO mon(i)
<12> 2650 c=LEN(tag$(i,j))+LEN(dat$(i,j))
<28> 2660 a=35:b=7
```

Listing KALENDER

```
<91> 2670 IF wt=0 OR tag(i,j)=1 THEN a=57:b=25
<77> 2680 LPRINT FNgem$(a);DEC$(j,"##.");wt$(wt);FNgem$
     (b);un.on$;
<16> 2690 LPRINT tag$(i,j);SPACE$(83-c);FNgem$(25);dat$
     (i,j);FNgem$(b);un.off$:LPRINT
<96> 2700 wt=wt+1:IF wt=7 THEN wt=0
< 1> 2710 NEXT
<36> 2720 IF tag(i,0)=1 THEN LPRINT:LPRINT FNgem$(21);f
     uze$
<72> 2730 LPRINT CHR$(12);
<10> 2740 NEXT
<45> 2750 GOTO 2100
<17> 2760 '
<20> 2770 ' Drucken (Einzelblatt)
<23> 2780 '
<76> 2790 tag=1:mo=mon:mon=1:GOSUB 1910:b=46:c=1
<84> 2800 PRINT cls$;FNloc$(10,10);:INPUT"Wenn Drucker
     bereit >RETURN< drücken ",a$
<42> 2810 LPRINT FNgem$(40);". Kalender für "land$(land
     );jahr
<44> 2820 LPRINT
<28> 2830 FOR i=1 TO 12:
<96> 2840 a$="":b$=""
<31> 2850 LPRINT FNgem$(40);" ";mon$(i)
<31> 2860 FOR j=1 TO mon(i)
<80> 2870 a=7:IF wt=0 OR tag(i,j)=1 THEN a=25:IF tag(i,
     j)=1 THEN GOSUB 3020:GOTO 2900
<90> 2880 IF i=3 AND INSTR(tag$(i,j),"Sommerzeit")>0 TH
     EN a$(c)=a$(c)+DEC$(j,"###.")+" 3. Sommerzeit":GOS
     UB 3030
<25> 2890 IF i=9 AND INSTR(tag$(i,j),"Winterzeit")>0 TH
     EN a$(c)=a$(c)+DEC$(j,"###.")+" 9. Winterzeit":GOS
     UB 3030
<45> 2900 a$=a$+FNgem$(a)+" "+wt$(wt)
<33> 2910 b$=b$+FNgem$(a)+" "+DEC$(j,"##")
< 7> 2920 wt=wt+1:IF wt=7 THEN wt=0
<11> 2930 NEXT
<59> 2940 LPRINT a$;zei.8$:LPRINT b$;zei.6$:LPRINT
<17> 2950 NEXT
<91> 2960 LPRINT " ";FNgem$(41)"Feiertage in";jahr;:LPR
     INT FNgem$(25)
<62> 2970 FOR i=1 TO 10:LPRINT a$(i):a$(i)="":NEXT:LPRI
     NT CHR$(12);
<58> 2980 GOTO 2100
<30> 2990 '
<85> 3000 ' Druckzeile erstellen
<88> 3010 '
<66> 3020 a$(c)=a$(c)+DEC$(j,"###.")+DEC$(i,"##.")+tag$
     (i,j)
<34> 3030 a$(c)=a$(c)+SPACE$(b-LEN(a$(c)))
<54> 3040 c=c+1:IF c>10 THEN c=1:b=92
<95> 3050 RETURN
< 4> 3060 '
< 7> 3070 ' + 1 Jahr
<10> 3080 '
<17> 3090 ERASE tag$,tag,dat$:jahr=jahr+1:mo=mon:IF jah
     r>2100 THEN jahr=1600:GOTO 1050 ELSE 1050
<87> 3100 '
<90> 3110 ' - 1 Jahr
<93> 3120 '
< 6> 3130 ERASE tag$,tag,dat$:jahr=jahr-1:mo=mon:IF jah
     r<1600 THEN jahr=2100:GOTO 1050 ELSE 1050
< 0> 3140 '
< 3> 3150 ' Programm beenden
< 6> 3160 '
<80> 3170 PRINT FNwin$(1,1,32,90);bildumon$;cls$
<81> 3180 WIDTH LPRINT 80
<55> 3190 LPRINT pica$+FNzese$(70)+FNleze$(3)+zei.6$+ei
     nblatt$+pendoff$+default$+ptr.reset$+CHR$(13);
<72> 3200 END
<92> 3210 '
<95> 3220 ' Daten für Wochentage
<98> 3230 '
<83> 3240 DATA So,Mo,Di,Mi,Do,Fr,Sa
< 5> 3250 '
< 8> 3260 ' Daten für Monate
<11> 3270 '
<26> 3280 DATA Januar,31,Februar,28,März,31,April,30,Ma
     i,31,Juni,30
<82> 3290 DATA Juli,31,August,31,September,30,Oktober,3
     1,November,30,Dezember,31
<91> 3300 '
<94> 3310 ' Namen der Länder
<97> 3320 '
<87> 3330 DATA Baden-Württemberg,Bayern,Berlin-West,Bre
     men,Hamburg,Hessen
<32> 3340 DATA Niedersachsen,Nordrhein-Westfalen,Rheinl
     and-Pfalz,Saarland
<25> 3350 DATA Schleswig-Holstein
<10> 3360 '
<13> 3370 ' Daten für Feiertage, die sich auf das Oster
     fest beziehen
<16> 3380 '
<19> 3390 ' 1. Spalte = anzahl der Tage vor oder nach O
     stern
<93> 3400 ' 2. Spalte = Kennzeichnung der Feiertage in
     den Ländern
<71> 3410 '... 0 = kein Feiertag
```

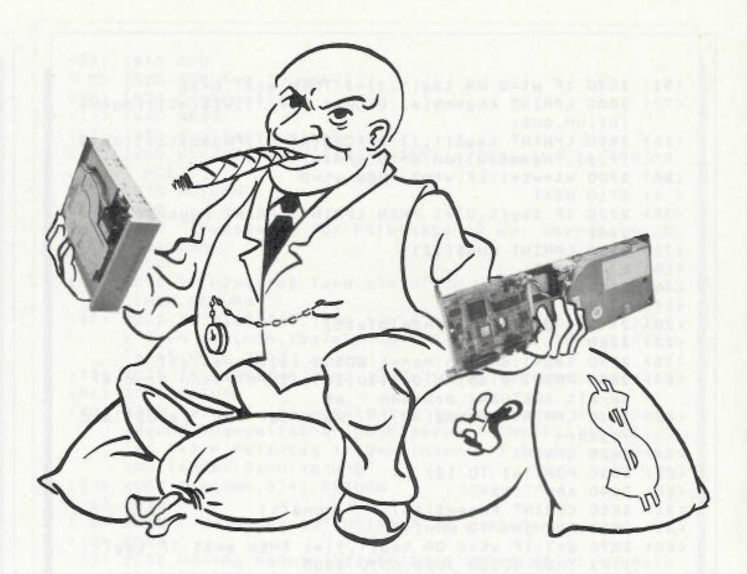
Listing KALENDER

```
<92> 3420 '... e = in Gemeinden mit überwiegend evangel
     ischer Bevölkerung
<94> 3430 '... k = in Gemeinden mit überwiegend katholi
     scher Bevölkerung
<83> 3440 '... x = gesetzlicher Feiertag
< 9> 3450 '
<41> 3460 DATA - 2,xxxxxxxxxxx,Karfreitag
<74> 3470 DATA.. 0,xxxxxxxxxxx,Ostersonntag
<88> 3480 DATA.. 1,xxxxxxxxxxx,Ostermontag
<56> 3490 DATA. 39,xxxxxxxxxxx,Christi Himmelfahrt
<49> 3500 DATA. 49,xxxxxxxxxxx,Pfingstsonntag
<10> 3510 DATA. 50,xxxxxxxxxxx,Pfingstmontag
<24> 3520 DATA. 60,xk000x0xxx0,Fronleichnam
<11> 3530 DATA -52, ,Altweiber-Fastnacht
<80> 3540 DATA -48, ,Rosenmontag
<65> 3550 DATA -47, ,Fastnacht
<73> 3560 DATA -46, ,Aschermittwoch
< 0> 3570 DATA - 7, ,Palmsonntag
<71> 3580 DATA - 3, ,Gründonnerstag
<95> 3590 DATA. 56, ,Dreifaltigkeitsfest
<93> 3600 DATA ,ENDE,
< 1> 3610 '
< 4> 3620 ' Daten für Feiertage,die sich auf das Kirche
     njahr beziehen
< 7> 3630 '
<10> 3640 ' 1. Spalte = Anzahl der Tage nach dem Volkst
     rauertag
<13> 3650 ' 2. Spalte = Kennzeichnung der Feiertage
<16> 3660 '
< 1> 3670 DATA. 3,xexxxxxxxxx,Buß- und Bettag
<15> 3680 DATA. 0, ,Volkstrauertag
<61> 3690 DATA. 7, ,Totensonntag
<46> 3700 DATA 14, ,1. Advent
<40> 3710 DATA 21, ,2. Advent
<44> 3720 DATA 28, ,3. Advent
<38> 3730 DATA 35, ,4. Advent
< 8> 3740 DATA ,ENDE,
<15> 3750 '
<18> 3760 ' Daten für sonstige Feiertage
<21> 3770 '
<24> 3780 ' 1. Spalte = Monat
<27> 3790 ' 2. Spalte = Tag
< 2> 3800 ' 3. Spalte = Kennzeichnung der Feiertage
< 5> 3810 '
<59> 3820 DATA. 1, 1,xxxxxxxxxxx,Neujahr
< 7> 3830 DATA. 1, 6,xx000000000,Heilige Drei Könige
<28> 3840 DATA. 5, 1,xxxxxxxxxxx,Maifeiertag
<29> 3850 DATA. 6,17,xxxxxxxxxxx,Tag der dt. Einheit
<67> 3860 DATA. 8,15,0k0000000x0,Maria Himmelfahrt
<59> 3870 DATA 11, 1,xk00000xxx0,Allerheiligen
<58> 3880 DATA 12,25,xxxxxxxxxxx,1. Weihnachtstag
< 6> 3890 DATA 12,26,xxxxxxxxxxx,2. Weihnachtstag
<87> 3900 DATA. 2,14, ,Valentinstag
<80> 3910 DATA. 5, 5, ,Europatag
<78> 3920 DATA. 5,11, ,Eisheiligen (11-13 Nord-D)
<47> 3930 DATA. 5,12, ,Eisheiligen (12-14 Süd-D)
<51> 3940 DATA. 5,15, ,Kalte Sophie
<72> 3950 DATA. 6,24, ,Johannisfest
<64> 3960 DATA. 6,27, ,Siebenschläfer
<20> 3970 DATA 10,31, ,Reformationtag
<71> 3980 DATA 11, 2, ,Allerseelen
<20> 3990 DATA 11,11, ,Martinstag
<77> 4000 DATA 12, 6, ,Sankt Nikolaus
< 6> 4010 DATA 12,24, ,Heiligabend
<26> 4020 DATA 12,31, ,Silvester
<56> 4030 DATA ,,ENDE,
<98> 4040 '
< 2> 4050 ' Daten für bewegliche Feiertage. -. Umstellu
     ng der Uhren auf Sommer- Winterzeit
< 5> 4060 '
< 8> 4070 ' 1. Spalte = Monat
<11> 4080 ' 2. Spalte = größtes mögliches Datum für die
     sen Tag
<14> 4090 ' 3. Spalte = Name des Tages
<88> 4100 '
<81> 4110 DATA. 3,31,Sommerzeit
<74> 4120 DATA. 5,14,Muttertag
<65> 4130 DATA. 9,30,Winterzeit
<13> 4140 DATA 10, 7,Erntedankfest
<28> 4150 DATA , ,ENDE
< 7> 4160 '
<10> 4170 ' Persönliche Daten. z.B. Geburtstage usw.
<13> 4180 '
<16> 4190 '
<90> 4200 ' DATA. Monat , Tag , Name
<93> 4210 '
<74> 4220 DATA. 1,13,Yvonne
<81> 4230 DATA. 2,18,Konstanze
<49> 4240 DATA. 3, 4,Kasimir
<34> 4250 DATA. 4,30,Rosamunde
<77> 4260 DATA. 5,30,Ferdinand
< 1> 4270 DATA. 7,24,Luise
<36> 4280 DATA. 9, 1,Ruth - Verena
<64> 4290 DATA. 9,13,Tobias
<89> 4300 DATA 10,23,Severin
< 3> 4310 DATA 11,24,Flora
<56> 4320 DATA 12,22,Jutta
< 7> 4330 DATA ,,ENDE
```

Listing KALENDER

Datenmillionär

Festplatten und der PC1512



Nach einer gewissen Zeit der Arbeit mit seinem PC wird wohl bei jedem der Wunsch nach einer Festplatte aufgetreten sein. Geht man fröhlich in einen Computer-Shop und möchte sich eine Festplatte kaufen, fragt der Händler meist nach dem Rechnertyp. Auf die Antwort Amstrad 1512 hört man dann oft: "Öh, ja, da kann man doch gar keine Platte anschließen, oder doch?" oder: "Ja, da geht nur eine Drivecard und keine normale Platte.". Auf bohrende Fragen, warum das denn angeblich so sei, kommt die Antwort: "Ja, das Netzteil ist soooo schwach, und der Rechner ist überhaupt nicht kompatibel."

Diese Antworten zählen größtenteils zu den Mythen der Computerwelt und haben ihre Ursachen darin, daß der Händler den Rechner nicht richtig kennt und die Meinung mancher 'Fachleute' aus 'Fachzeitschriften' blind wiederholt. Um einige irrige Meinungen über den Anschluß von Festplatten am PC 1512 zu berichtigen, wurde dieser Artikel geschrieben.

Festplatten und Drivecards

Zuerst eine kurze Erklärung zu den Unterschieden zwischen einer normalen Festplatte und einer Drivecard. Während bei der Festplatte das eigentliche Laufwerk und der Festplattencontroller getrennte Einheiten sind, sind Controller und 3,5-Zoll-Laufwerk bei einer Drivecard zusammen auf einem Einsteckrahmen aufgebaut. Dieser Einsteckrahmen wird 'einfach' in einen freien Slot des Rechners gesteckt. So einfach dadurch auch der Einbau wird (Kiste auf, Karte rein, Kiste zu und das Ding läuft), sollte man daran denken, daß durch die etwas 'dickeren' (volumenmäßig gesprochen) Laufwerksteile auf der Drivecard meist 1,5 bis 2, teilweise sogar 3 Steckplätze belegt sind. Es gibt zwar extrem flache Drivecards, diese sind aber auch meist einige hundert bis tausend Mark teurer als normale. Da wir schon bei Preisen sind, eine Drivecard ist, bei gleicher Zugriffsgeschwindigkeit und gleichen Speicherplatz, teurer als ein gleichartiges Festplattenlaufwerk. Natürlich gibt es auch bei den normalen Laufwerken viele Unterschiede. Der sofort ins Auge fallende ist natürlich die Größe. Es gibt inzwischen die altbekannten 5,25-Zoll-Laufwerke (etwa so groß wie ein Diskettenlaufwerk für 5,25-Zoll-Disketten) und die immer mehr in Mode kommenden 3,5-Zoll-Laufwerke (entschieden kleiner und auch auf Drivecards zu finden). Sonstige Unterschiede liegen meistens in der Art des Controllers und der damit zusammenhängenden Elektronik (ST506, ESDI, SCSI usw.) und interessieren hier nur am Rande.

Drivecard oder Festplatte?

Was soll man nun nehmen? Zuerst: Man kann prinzipiell alle Festplattenlaufwerke am PC 1512 anschließen. Das größte Problem dürfte hierbei wohl in der Finanzkraft des einzelnen liegen. Dann wäre als nächstes zu überlegen, ob es eine Drivecard oder ein herkömmliches Plattenlaufwerk sein soll. Beim PC1512/1640 stehen nur drei Steckplätze zur Verfügung. Wer eine Drivecard einsetzt, verliert meist zwei dieser Steckplätze. Andererseits, wer schon zwei Diskettenlaufwerke hat, möchte ungern auf eines davon verzichten. Nun, wer genau weiß, daß er keine weiteren Steckplätze benötigt, kann ruhig eine Drivecard nehmen. Wer noch mehr Karten benötigt und trotzdem nicht auf sein zweites Diskettenlaufwerk verzichten möchte, dem sei gesagt – wenn es sich um zwei gleichartige Laufwerke handelt – man braucht sehr selten ein zweites Diskettenlaufwerk, wenn man eine Festplatte hat. Handelt es sich aber um verschiedenartige Diskettenlaufwerke (3,5 und 5,25 Zoll) gibt es zwei Möglichkeiten:

1. doch eine Drivecard,
2. ein Diskettenlaufwerk extern anschließen.

Bei Tip für solche, deren Augen gerne mit angenehmen visuellen Reizen versorgt werden: Ein externes Laufwerk mit Mahagoni-Furnier-Gehäuse sieht sehr gut aus und paßt vielleicht sogar zur Schrankwand.

Anschlußprobleme

Mythen haben auch ein Körnchen Wahrheit in sich. Es gibt wirklich zwei Probleme (bei einer Festplatte sind es drei). Das erste ist die Stromversorgung. Das Netzteil des Rechners ist ein wenig schwach auf der Brust. Hat man aber nur die Grundausstattung im Rechner (ein oder zwei Laufwerke, keine sonstigen externen Karten) gibt es mit keiner Platte Probleme. Wer darauf achtet, ein Laufwerk mit niedrigem Anlaufstrom zu bekommen, bekommt auch mit mehr Ausstattung keine Probleme (mein eigener Amstrad 1512 läuft seit einigen Jahren problemlos mit Festplatte, 3,25- und 5,25-Zoll-Laufwerken, Ventilator, Speichererweiterung, höherem Takt, V30-Prozessor usw.).

Problem Nummer zwei ist die Wärmeentwicklung eines Festplattenlaufwerks. Bei einer Drivecard kann dies dadurch gelöst werden, daß man die hintere Abdeckung offenläßt. Bei einer Festplatte sollte (muß) man sich einen Ventilator kaufen und einbauen (bei neueren Modellen, erkennbar an seitlichen Lüftungsschlitzen, sind schon Befestigungen für einen Ventilator vorgesehen).

Das letzte Problem tritt nur bei einem Festplattenlaufwerk auf. Da Amstrad einiges anders machen mußte, ist der mechanische Einbau etwas für Tüftler (es gibt aber auch Einbausets für Amstrad-Rechner zu kaufen). Wer auf die Idee kommen sollte, den Einbaukasten seines eventuell überflüssigen Diskettenlaufwerkes zu verwenden, sei vor den dadurch entstehenden thermischen Problemen gewarnt. Ein Festplatten-

laufwerk erzeugt sehr viel Wärme und mag es überhaupt nicht, wenn diese Wärme nicht vernünftig abgeführt wird. Eine Möglichkeit ist, nur die Seitenteile des Gehäusekastens zu verwenden.

Leider stimmen aber die meisten Bohrungen nicht mit denen der Festplatte überein (welchen Bastler stört so etwas aber?). Vor dem Einbau sollte kontrolliert werden, ob die Jumpereinstellungen auf der Controllerkarte und dem Laufwerk stimmen. (Sie haben doch daran gedacht, von Ihrem Händler das Controllerhandbuch zu verlangen?)

Nach dem mechanischen Einbau

Befindet sich die Platte glücklich im Rechner, hat man das größte (Einbau-) Problem hinter sich. Was jetzt noch kommt, ist mehr oder weniger Kleinkram. Das erste, was nach dem Einbau folgt, ist die sogenannte physikalische Vorformatierung der Platte. Die Platten sind zwar meist vorformatiert, jedoch mit Sektorinterleave-Werten (Skew) für den guten, alten IBM-PC. Bevor Sie sich jetzt fragen, was denn ein Sektorinterleave ist, hier die kurze Erklärung:

Eine Festplatte liefert die Daten sehr schnell. Dadurch passiert es oft, daß, wenn der Rechner den nächsten Sektor anfordert, dieser gerade am Schreib/Lesekopf vorbeigerauscht ist. Um dies zu verhindern, wird die Reihenfolge der Sektoren geändert (Sektorinterleave). Wenn der nächste (logische) Sektor für den Rechner erst der übernächste (Skew == 2) oder der drittnächste (Skew == 3) physikalische Sektor ist, so kommt die Anforderung gerade noch rechtzeitig.

Für einen normalen PC1512/1640 (mit normalem Controller, einem sogenannten ST506, kein ESDI, SCSI oder ähnlichem) ist ein Interleavefaktor von 4 recht gut. Hat man einen V30-Prozessor eingebaut oder die Taktfrequenz erhöht, empfiehlt sich ein Wert von 3. Doch wie formatiert man seine Platte physikalisch?

Physikalisches...

Dazu gibt es zwei Möglichkeiten. Zu den meisten Festplattenlaufwerken gehören auch Installationshilfen wie Ontrack's "DiskManager" oder ähnliches. Hat man ein Installationsprogramm zu seiner Platte mitbekommen, so geschieht die Vorformatierung mit Hilfe dieses Programms. Hat man kein Installationsprogramm, muß man den DOS-Debugger benutzen. Auf jeder Controllerkarte befindet sich eine BIOS-Erweiterung. Sie dient zur Steuerung der Platte, enthält aber auch eine Routine zur physikalischen Formatierung. Der Start der Routine läuft etwa (je nach Controller, näheres steht im Controllerhandbuch) so ab:

A>debug -rcs :C800 -u6 (ausgegebene Befehl muß mit JMP beginnen) -g=6

Die Routine beginnt meist mit einer Meldung, daß sie eine Formatierroutine ist und sämtliche Daten unwiderbringlich löscht. Es werden dann verschiedene Daten abgefragt, unter anderem auch der Sektorskew. Hat man alles eingegeben, beginnt die physikalische Formatierung der Platte. Der nächste Schritt ist die Einteilung der Platte in Partitionen mit Hilfe von FDISK (siehe DOS-Handbuch).

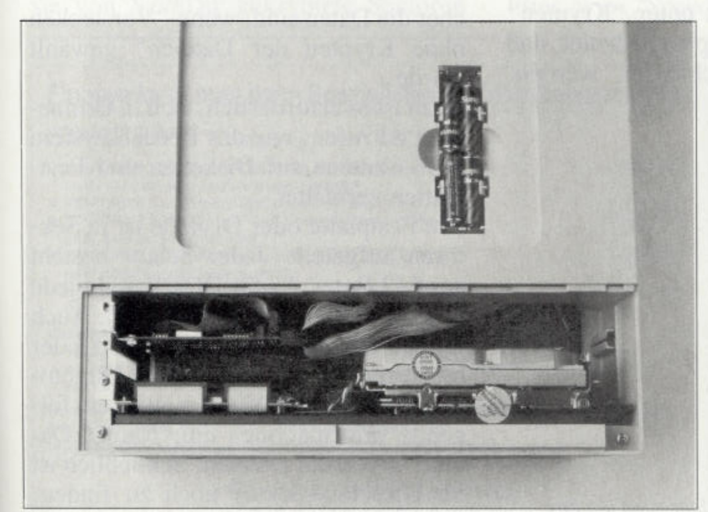
...und logisches Formatieren

Hat man dies auch hinter sich gebracht, muß man die Platte noch logisch formatieren. Dies geschieht mit Hilfe des normalen DOS-FORMAT-Programms. Im Unterschied zum Formatieren einer Diskette mit FORMAT wird die Festplatte von FORMAT nur nach defekten Sektoren abgesucht und der Bootsektor, die FAT und das Haupt-Directory neu geschrieben (deshalb nennt sich das auch logisches Formatieren). Hat man diese ganze Prozedur hinter sich, ist man Besitzer von einigen Speichermillionen, die aber mit der Zeit immer kleiner erscheinen.

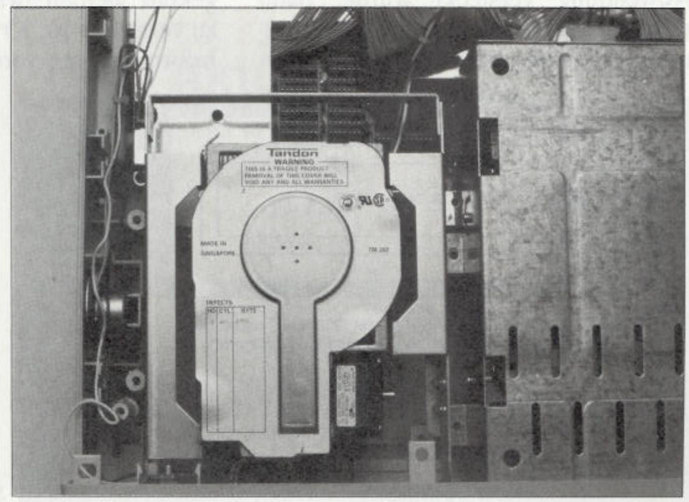
Der Schein trügt

Denn nach den ersten Installationen der wichtigsten Anwenderprogramme zeigt sich schnell, warum es der sprichwörtliche Millionär so "schwär" hat, denn: Wer viel hat, der gibt erstens viel aus und will zweitens immer mehr. Denken Sie also daran, daß auch 20 MByte sehr schnell bespielt sind. Die Hochstimmung nach Ansehen der ersten Meldung über freien Speicherplatz schlägt nur allzu schnell ins Gegenteil um. Sparsamkeit ist auch bei einer Festplatte angesagt.

(Robert Haas/jf)



Eine Filecard ist flexibel, nimmt aber meist mehr als einen Steckplatz für sich in Anspruch



Die Vorteile einer Festplatte liegen vor allem im günstigeren Preis

SAVEDIR

Schützt das Directory!

Eine der Schwächen von MS-DOS ist sicherlich die fehlende Möglichkeit, Daten durch ein Passwort vor unberechtigten Zugriffen zu schützen. Gerade dort, wo mehrere Personen Zugang zu Rechnern mit Festplatte haben, ist dieser Wunsch nicht so abwegig. Disketten lassen sich mitnehmen oder einschließen, aber Festplatten verbleiben im Gerät und sind somit gefährdet. Hier gibt es nun Abhilfe: die Software-Lösung SaveDir.

Der Schutz von SaveDir läßt sich auch mit Utilities (z.B. Norton Utilities, PC-Tools usw.) nicht aufbrechen. Die von SaveDir geschützten Programme und Daten sind wirklich nur dem berechtigten Benutzer zugänglich. SaveDir schützt extrem schnell: 10 MByte in wenigen Sekunden. In SaveDir sind an den entscheidenden Stellen laufzeitoptimierte Assembler-Routinen am Werk. Diese machen es möglich, daß SaveDir 10 MByte auf der Festplatte eines AT-Computers (6 MHz, 28 msec-Platte) in weniger als 8 Sekunden schützt beziehungsweise hervorholt! Nicht nur die extrem gute Laufzeit ist ein wichtiger Punkt. Auch das Konzept, mit dem SaveDir die Daten schützt, macht es zu einer echten Hilfe. Nachdem der berechtigte Benutzer die Programme und Daten nach Eingabe seines Passwortes hervorgeholt hat, sind diese wieder benutzbar. Man kann also ohne jede Einschränkung damit arbeiten.

Das Programm läßt sich sehr einfach bedienen und installieren: Man ruft es auf, und das Programm fragt zuerst das Passwort ab. Danach zeigt SaveDir dem Benutzer, welcher Vorgang an der Reihe ist: "Versteckenʺ oder "Hervorholenʺ. Wenn bei der letzten SaveDir-Anwendung versteckt wurde, dann wird bei der aktuellen Sitzung automatisch "Hervorholenʺ angeboten. Man kann auch als Anfänger nichts falsch machen.

Auch das Installationsprogramm hat eine einfache und übersichtliche Struktur. Man gibt an, welches Directory geschützt werden soll. In wenigen Minuten ist SaveDir installiert.

SaveDir ist flexibel und läuft auf allen Festplatten- und Diskettenformaten. Da sich das Programm des Betriebssystems bedient, um zu schützen, ist es unabhängig von der Hardware. Es arbeitet auf Disketten (5,25 oder 3,5 Zoll) und auf Festplatten. Selbst wenn "großeʺ Festplatten mit einer Sektor-Größe von mehr als 512 Byte verwendet werden, gibt es keinerlei Probleme. Das Programm benötigt allein MS-DOS-Versionen von 2.0 an aufwärts.

Verstecken mit oder ohne Krypten?

Im Menü zum Verstecken des Directories bietet SaveDir dem Benutzer zwei Sicherheitsstufen an:

1. Verstecken ohne Krypten der Dateien
2. Verstecken mit Krypten der Dateien

Es ist wichtig, daß man weiß, was der Unterschied zwischen den beiden Wahlmöglichkeiten ist. Zunächst soll geklärt werden, was unter "Kryptenʺ zu verstehen ist. Krypten bedeutet, daß Informationen verschlüsselt werden. Verschlüsselte Informationen sind Datenschrott, eine Ansammlung sinnlos erscheinender Bytes. Erst durch das Entkrypten wird der Zustand vor dem Krypten wiederhergestellt. Das Entkrypten kann bei SaveDir nur mit dem korrekten Passwort in Gang gebracht werden.

Ein sofort bemerkbarer Unterschied zwischen "Verstecken ohne Krypten der Dateienʺ und "Verstecken mit Krypten der Dateienʺ ist, daß beim Verstecken mit Krypten SaveDir wesentlich länger braucht. Das liegt ganz einfach daran, daß SaveDir in diesem Modus Sektor für Sektor lesen, krypten und dann wieder zurückschreiben muß.

Das Verschlüsseln selbst dauert nicht lange. Hier sind laufzeitoptimierte Assembler-Routinen am Werk, die schnell und sicher kodieren. Das Lesen und Zurückschreiben dauert da schon länger. Es gibt keine Möglichkeit, diesen Prozeß abzukürzen. Wenn gekryptet werden muß, dann muß nun einmal Sektor für Sektor von der Festplatte in den Arbeitsspeicher gelesen und von dort nach getaner Arbeit wieder zurück auf die Festplatte geschrieben werden.

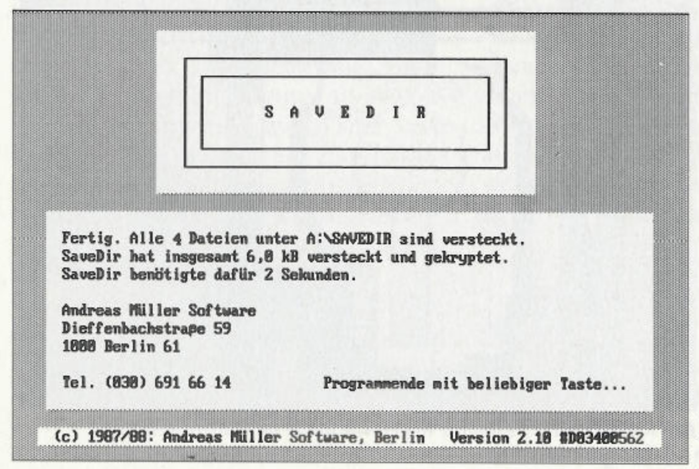
Unabhängig davon, ob man sich für "Verstecken ohne Krypten der Dateienʺ oder "Verstecken mit Krypten der Dateienʺ entscheidet: versteckt wird in beiden Fällen. Jedesmal werden auch die Zugriffsinformationen auf das versteckte Directory verschlüsselt. Beim Krypten wird aber zusätzlich noch jeder versteckte Daten-Sektor verschlüsselt.

Die Unterschiede beim "Versteckenʺ

Im folgenden wird aus technischer Sicht der Unterschied erklärt, wie sicher die Daten sind, wenn "Verstecken ohne Krypten der Dateienʺ gewählt wurde.

Dazu ist es erforderlich, sich in Erinnerung zu rufen, wie das Betriebssystem DOS Dateien auf Disketten und Festplatten verwaltet.

Die Festplatte oder Diskette ist in Sektoren aufgeteilt. Jeder Sektor besteht aus 512 Bytes. Unter DOS besteht jede Datei aus einer Anzahl Sektoren. Auch Directories bestehen aus einem oder mehreren Sektoren. In einem Directory-Sektor trägt DOS unter anderem folgende Informationen ein: Name, Datum, Größe der Dateien. Schließlich ist im Directory-Sektor noch zu finden, wo auf der Festplatte die Dateien jeweils beginnen. In einem anderen Sek-



Schutz rund um Ihre Datenbestände bietet Ihnen SaveDir



tor (genannt FAT: File Allocation Table) verwaltet DOS die Information, welche Sektoren noch zu dieser Datei gehören. Die einzelnen Sektoren der Datei dürfen weit über die Festplatte verstreut stehen. Über die FAT gelingt es DOS immer, die einzelnen Sektoren zusammenzusuchen.

SaveDir setzt nun genau an diesem Punkt an. Es verschlüsselt die Directory-Sektoren und entfernt aus der FAT die Information, welche Sektoren in welcher Reihenfolge den Dateien zugeordnet sind. SaveDir verschlüsselt also sämtliche Zugriffsinformationen. Dadurch kann man auch mit Utilities nicht in die von SaveDir geschützten Directories eindringen.

Von den Directory-Sektoren sind die Daten-Sektoren zu unterscheiden. Sie beinhalten die eigentlichen Datei-Inhalte.

Mit Sicherheitsstufe 1, “Verstecken ohne Krypten der Dateien“, werden die Daten-Sektoren unverändert auf der Festplatte belassen. Mit speziellen Utilities (z.B. Norton Utilities, PC-Tools) ist es möglich, Sektoren unabhängig von der Zuordnung des Sektors zu einer Datei zu lesen.

Damit könnten also auch Sektoren aus Bereichen von einer Fachkraft gelesen werden, die von SaveDir bzw. der SaveDir-Demo versteckt wurden. Die Fachkraft hat keine Chance zu erfahren, wie der Dateiname war, zu welcher der gelesene Sektor gehört hat, oder ob nach oder vor diesem Sektor noch weitere Sektoren sind, die zusammen einmal eine Datei waren.

Mit Sicherheitsstufe 2, “Verstecken mit Krypten der Dateien“, werden zusätzlich zu den Directory-Sektoren auch die Daten-Sektoren verschlüsselt. Auch eine noch so erfahrene und/oder intelligente Fachkraft sieht sich einer unlösbaren Aufgabe gegenüber, wenn sie von SaveDir gekryptete Sektoren auswerten soll, da sich dem unberechtigten “Einseher“ lediglich ein unzusammenhängender “Datenbrei“ offenbahrt. Der gekryptete Sektor hat mit dem ursprünglichen Sektor nichts mehr gemein. Alle möglichen Bytes stehen wild durcheinander, Text ist absolut nicht mehr auszumachen. Nur SaveDir kann, nach korrekter Eingabe des Passwortes, den Zustand vor dem Verstecken wiederherstellen.

Wenn niemand erfahren soll, was im “SaveDir-Directory“ auf der Festplatte steht, wenn verhindert werden soll, daß unerlaubt Programme von der Festplatte herunterkopiert werden, oder wenn man erreichen will, daß ein unerfahrener DOS-Benutzer die wertvollen Programme nicht versehentlich zerstören kann, dann genügt es, “Verstecken ohne Krypten der Dateien“ anzuwählen.

Müssen aber auch die Datei-Inhalte verschlüsselt werden, dann sollte “Verstecken mit Krypten der Dateien“ gewählt werden. Danach sind die Daten sicher.

Verträglichkeit von SaveDir mit anderen Programmen

Welche Wahl die richtige ist, muß der Anwender selbst anhand seiner Sicherheitsanforderungen entscheiden.

Wenn Daten verschlüsselt werden sollen, dann ist angeraten, diese wichtigen Daten in einem eigenen Directory unterzubringen und speziell für dieses Directory eine SaveDir-Installation vorzunehmen. So kann vermieden werden, daß SaveDir Dateien bearbeitet, die gar nicht verschlüsselt werden müssen. Das Programm bietet damit für jeden Sicherheitsanspruch die richtige Lösung.

SaveDir kann auch zusammen mit Utilities wie “Speed Disk“ der Norton Utilities 4.0 Advanced Edition, “COMPRESS“ von den PC-Tools und “Disk Optimizer“ betrieben werden. Diese Utilities stellen keine Gefahr für die versteckten Programme und Daten dar. Sie organisieren die Anordnung der Dateien auf der Festplatte so, daß das Betriebssystem auf sie optimal schnell zugreifen kann. In Zusammenhang mit SaveDir treten dabei keinerlei Schwierigkeiten auf. Wenn ein Directory vor der Optimierung versteckt wurde, dann werden die in dem Directory enthaltenen Dateien allerdings nicht mit optimiert, denn auch diese Programme haben keinen Zugang zu dem versteckten Directory. Wird nach einem “Hervorholen“ des Directories optimiert, dann werden natürlich auch die jetzt zugänglichen Dateien des Directories mit allen anderen Dateien optimiert. Es bietet sich demnach an, vor der Arbeit mit SaveDir die Datenträger mittels Utility zu ordnen.

Einem guten Produkt eine gute Beurteilung

SaveDir hat bisher nur “gute Noten“ bekommen. Dieser Eindruck kann auch an dieser Stelle nur bestätigt werden. An diesem brauchbaren Utility stimmt einfach alles: Funktion, Anwender-Freundlichkeit (gilt auch für das Handbuch), Betriebssicherheit und Preis-Leistungsverhältnis.

Liefernachweis
Andreas Müller Software
Dieffenbachstraße 59
1000 Berlin 61
Telefon: (030) 691 66 14
Das Programm kostet rund 200 DM.

(Oliver Rosenbaum/jf)



Alle Macht den Mäusen!

Simulierte Tastendrücke mit UseYourMouse

Es gibt immer noch Programme, die eine angeschlossene Maus beharrlich ignorieren und auf einer Tastaturbedienung bestehen. Mit UseYourMouse kann die meisten dieser Programme Mores gelehrt werden.



UseYourMouse, das nun in der Version 2.0 vorliegt, ist ein kleines residentes Programm, mit dem Mausbewegungen in simulierte Tastendrücke umgewandelt werden. Dadurch ist es möglich, vielen bislang nur über die Tastatur zu bedienenden Programmen nun auch die Maus schmackhaft zu machen. Jedoch nur in beschränktem Umfang, da für die Maus nur sechs bis sieben Aktionsmöglichkeiten gegeben sind, nämlich vier Bewegungsrichtungen, die meist den Cursortasten zugeordnet werden, und bis zu drei Maustasten, die mit anderen wichtigen Tasten wie ENTER oder ESC belegt werden können. Mehr ist beim besten Willen nicht drin und so kann es sein, daß die Verwendung der Maus nicht immer ein ergonomischer Fortschritt ist.

Das jeweilige Programm wird durch die Emulation nicht wirklich mausorientiert, der Griff nach rechtsaußen für einige Cursorbewegungen wird zum reinen Training für die Armmuskulatur (was ja ein willkommener Nebeneffekt sein kann), wenn die nächsten Befehle wieder über die Tastatur erfolgen müssen.

Über den Sinn einer solchen Maßnahme kann man durchaus geteilter Meinung sein, seine technische Aufgabe erledigt UseYourMouse jedenfalls zufriedenstellend. Das Spektrum der zum Test herangezogenen Software war breit, bei den Anwendungen gab es keinerlei Probleme. (Getestet wurden unter anderem Context, WordStar, dBase und Star Planer.)

Probleme bei Spielen

Kritischer wird es bei Spielen, da diese eher direkt auf die Tastatur zugreifen und dadurch die Emulation wirkungslos ist. Hier muß dann experimentiert werden. Als kleines Extra ist eine spezielle Installation von UYM für Sierra-Adventures enthalten, die sogar praktikabler ist als die von Sierra in den neueren Spielen angebotene Mausunterstützung.

Im Normalfall arbeitet UYM mit allen Microsoft- und Mouse-Systems-kompatiblen Mäusen und deren Treiber zusammen, durch den letzteren Aspekt ist er MausAll, dem nächsten Konkurrenten, überlegen, wenngleich UYM doppelt 'soviel' Platz, nämlich 1200 Byte, im Speicher belegt. Zum Test standen eine serielle Noname-Maus mit Originaltreiber von Microsoft sowie ein Trackball von Chicony, der beide Standards liefert, zur Verfügung. In keiner Kombination kam es zu Problemen, obwohl der Trackball eher zu Inkompatibilitäten neigt. Relativ wenig bringt UseYourMouse allerdings bei den Rechnern PC 1512 und 1640, hier werden die Mausbewegungen schon von Haus aus in Cursoraktionen umgesetzt. Die optimale Tastenkombination muß auf die jeweilige Anwendung zugeschnitten werden, hier leistet das übersichtliche Setup-Programm Hilfe. Mit ihm können beliebige Varianten bequem erstellt werden, wobei jeder der Mausbewegungen eine Taste zugeordnet werden kann. Zwar sind hier Kombinationen wie CTRL-C möglich, nicht jedoch Tastenketten wie CTRL-KB, die für manche Anwendungen nützlich wären. Mit einem Zusatzprogramm kann die Empfindlichkeit der Maus eingestellt werden, hier ist aber leider keine getrennte Einstellung für vertikale und horizontale Sensitivität möglich. Abgerundet wird das Paket durch ein Testprogramm für alle Mausfunktionen und eine deutsche Anleitung.

Alles in allem erhält man mit UseYourMouse eine solide Lösung, die in vielen Fällen hilfreich sein kann. Angesichts des niedrigen Preises kann man sich das Programm getrost zulegen, auch wenn die Erfolge manchmal nicht die gewünschten sind...

(Michael Anton/jf)



Die Bedieneroberfläche von UseYourMouse: wenig für das Auge, aber effektiv

Info:
Programm: UseYourMouse 2.0
Hardware: IBM/kompatible mit Microsoft-oder Mouse Systems-kompatibler Maus.
Hersteller: Martin Hepp Computertechnik, 6239 Eppstein.
Preis: 28 Mark



Wider die Datenmanipulation

Die versteckte AUTOEXEC.BAT-Datei



Es gibt findige Köpfe, die sich den lieben langen Tag damit beschäftigen, Datensicherheit und Schutz vor Fremdzugriffen zu ersinnen. Für den "normalen" PC-Besitzer sind diese Maßnahmen meist nicht so brauchbar, da sie entweder im Preis oder in der Wirkungsweise über das Maß der sinnvollen Nutzung gehen. Für diejenigen, die sich mit ein paar kurzen Handgriffen vor "bösen Buben" verteidigen wollen, gibt es meist schon kleine, aber sehr effektive Schutzmaßnahmen.

Besonders in Firmen, deren Mitarbeiter eigene Terminals bedienen oder in den eigenen vier Wänden, in denen Familienmitglieder gerne den ein oder anderen meist harmlosen Scherz mit dem Computernutzer treiben, ist der Umbau der AUTOEXEC.BAT-Datei ein beliebtes Spielfeld. Hier können verschiedene Kommentare oder Programmaufrufe untergebracht werden, die der Nutzer dann zwangsläufig über sich ergehen lassen muß. Dann schaltet man nichtsahnend den Computer an und sieht sich plötzlich mit irgendwelchem Unfug konfrontiert.

So nett oder lustig das auf der einen Seite sein kann, so böse kann solch eine kleine Veränderung auf der anderen Seite sein. In jedem Fall kostet die Wiederherstellung der alten Zustände Zeit, und in extremen Fällen ist es zusätzlich denkbar, daß ein Programm, welches auf der Festplatte untergebracht wurde und danach von der AUTOEXEC.BAT aufgerufen wird, unter Umständen mit einen Virus befleckt ist, was der "Täter" ja nicht wissen konnte.

Es ist daher ratsam, seine AUTOEXEC.BAT nicht offen auf der Festplatte liegenzulassen.

...und sie ward nicht mehr gesehn

Drei Schritte sollen uns helfen, den Computer zumindest vor dem Eingriff in besagte Startdatei zu immunisieren:

1. Änderung des Namens in der COMMAND.COM-Datei
2. Verstecken der so im Namen veränderten (ehemaligen) AUTOEXEC.BAT-Datei
3. Errichten einer Scheindatei mit Namen "AUTOEXEC.BAT"

Im Kommandointerpreter des MS-DOS-Betriebssystems (COMMAND.COM) befindet sich die Aufforderung an den Computer, nach dem Booten die Datei AUTOEXEC.BAT aufzurufen. Da der Dateiname im Klartext im Interpreter zu finden ist, bedarf es keiner großen Probleme, diesen Namen mit etwas Sorgsamkeit zu verändern. Wir benötigen dazu einen File-Editor (ein solcher ist in fast jeder Benutzeroberfläche enthalten). Nach dem Start dieses Programms laden wir den Kommandointerpreter COMMAND.COM. Mit ein wenig Geduld findet sich schnell der gesuchte Name AUTOEXEC.BAT. Diesen können wir nun verändern, wobei unbedingt darauf geachtet werden sollte, daß nur diese zwölf Stellen verwandelt werden sollten. Am besten wüten Sie nicht allzusehr im alten Namen herum. Eine Veränderung an einer einzigen Stelle reicht für einen effektiven Schutz meist schon aus. Füllen Sie auf keinen Fall bei einer etwaigen Wahl eines kürzeren Dateinamens die restlichen Stellen mit Leerzeichen aus.

Nach getaner Arbeit muß die Datei zurück auf den Datenträger geschrieben werden, dann können Sie den Editor verlassen.

Als zweites muß nun die alte AUTOEXEC.BAT-Datei mit dem neuen Namen versehen werden.

REN autoexec.bat <neuer Name>

Diese neue Datei könnte bei einem möglichen Datentäter bestimmte Verdachtsmomente aufkommen lassen. Wir wollen es nicht soweit kommen lassen und verstecken die Datei. Dazu muß das sogenannte Hidden-Flag der Datei gesetzt werden. Die einfachste Methode ist das Setzen mit Hilfe einer der gängigen Benutzeroberflächen. Gehen Sie dazu in das Attribut-Menü, rufen die neubenannte Datei auf und lassen Sie das Hidden-Flag setzen. Danach werden die neuen Attribute gesichert, und die "wahre" AUTOEXEC.BAT-Datei erscheint nicht mehr im Directory.

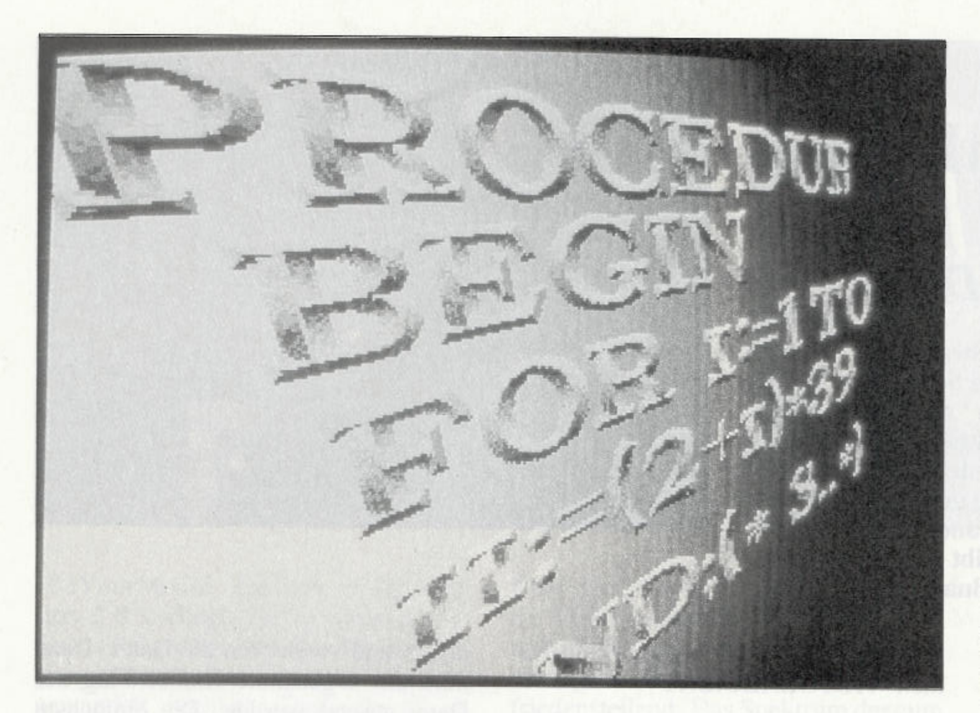
Der letzte Schritt ist nun das Legen einer falschen Fährte. Sollte jemand das Inhaltsverzeichnis Ihres Datenträgers durchsuchen und dabei keine AUTOEXEC.BAT-Datei finden, so wird er sicherlich schon bald stutzig werden. Nach dem Motto "Soll er/sie doch machen, was er/sie will, is mir doch wurscht!" errichten wir eine Datei namens AUTOEXEC.BAT und füllen sie mit allerlei plausiblen Befehlen. Der Bösewicht kann darin ändern, was er ändern will, Sie haben Ihre Ruhe vor derartigen Attentaten, und alle sind zufrieden.

Die Warnung darf nicht fehlen

Wie bei allen Änderungen an lebenswichtigen Dateien, sollten Sie auch in diesem Fall vorab Sicherheitskopien der zu ändernden Dateien anlegen.

Da es sich in unserem Fall um die Dateien COMMAND.COM und AUTOEXEC.BAT handelt, empfiehlt es sich, eine eigens zu diesem Zweck zubereitete Systemdiskette als Testobjekt zu verwenden. Wenn alles zu Ihrer Zufriedenheit läuft, kann die Festplatte immer noch umgebaut werden.

(jf)



Strukturierter Programmcode

Großgeschriebene Schlüsselworte in allen Programmiersprachen

Bei der Eingabe eines Programmtextes einer Programmiersprache in den Editor verwendet man im allgemeinen durchgängig die Kleinschreibweise, um bei der Eingabe schneller zu sein. Dadurch wird die optische Trennung von Befehlen und Variablen erschwert.
Das vorliegende Programm hilft dem gestreßten Programmierer etwas, indem nach Abschluß einer Entwicklungsphase einfach alle in einem Programmtext vorkommenden Schlüsselwörter in einem Durchgang durch ihre gemischt- (ClrScr) oder großgeschriebenen (PROCEDURE) Vorgaben ersetzt werden.

Dabei werden sowohl die Inhalte von Kommentaren als auch die in Anführungszeichen stehenden Teile von Zeichenketten nicht verändert, so daß z.B. das Wort 'in' eines Ausgabetextes nicht durch den großgeschriebenen Operator IN ersetzt wird.

Auch in Zeichenketten stehende Kommentarbegrenzer ('(*', '*)', '{', '}') oder in Kommentaren in Anführungszeichen stehende Zeichenketten bringen das Programm solange nicht durcheinander, wie es der Pascal-Compiler noch schafft, den Programmtext fehlerfrei zu compilieren.

Verfahrensweise

Zum Laufenlassen des Programms UPCASE wird außer der EXE-Datei noch die Datei UPCASE.DAT im aktuellen Verzeichnis benötigt. In dieser Datei sind die Vorlagen für die Schreibweise der Schlüsselwörter enthalten.

Wird die UPCASE.DAT vom Programm nicht im aktuellen Verzeichnis gefunden, bricht das Programm ab und setzt außerdem ERRORLEVEL auf den Wert 1, der dann in einer Batch-Datei abgefragt werden kann.

Außerdem wird auf dem Laufwerk, das die Quelldatei enthält, noch freier Speicherplatz in der Größe der umzuwandelnden Datei benötigt. Falls nicht genügend Speicherplatz vorhanden ist, bricht das Programm ebenfalls mit ERRORLEVEL ab.

Beim Aufruf von der DOS-Ebene kann direkt der Name der umzuwandelnden Programmdatei angegeben werden, wobei Pfadangaben zugelassen sind, z.B. 'UPCASE PROGRAMS \ meinprog.pas'. Wenn das Programm ohne Angabe eines Dateinamens aufgerufen wird, wird nach dem Namen der umzuwandelnden Datei im Dialog gefragt ('Umzuwandelnde Datei: __');

Wenn die angegebene Datei nicht gefunden wird, gibt das Programm eine Meldung aus ('Laufwerk: \ Pfad \ Dateiname.PAS nicht gefunden. Bitte neu eingeben.') und wartet auf die erneute Eingabe eines Dateinamens. Bei der Eingabe des Dateinamens kann die Dateiendung .PAS weggelassen werden, sie wird vom Programm automatisch angehängt, wenn im Programmnamen kein Punkt vorhanden ist.

Das Programm erwartet ein syntaktisch richtiges PASCAL-Programm als Eingabedatei, das heißt, beim Compilieren dürfen keine Fehler aufgetreten sein.

Erzeugt wird eine Textdatei in dem angegebenen Verzeichnis mit dem gleichen Namen wie die angegebene Datei, aber mit der Endung .UPC. In dieser Datei sind dann alle Schlüsselwörter je nach Vorgabe gemischt oder großgeschrieben. Anschließend sollte man die eventuell vorhandene *.BAK- Datei löschen, die alte *.PAS-Datei in *.BAK und die *.UPC-Datei in *.PAS umbenennen.

Hier ein Beispiel für eine Batch-Datei, die das Umbenennen komfortabler macht. Diese Batch-Datei wird im gleichen Verzeichnis wie der Turbo-Pascal-Compiler abgelegt und greift auf die Pascal-Programme zu, die im Unterverzeichnis PROGRAMS abgelegt sind.

```
IF ERRORLEVEL 1 GOTO ENDE
IF T%1 = = T GOTO ENDE
DEL PROGRAMS \ %1.BAK
REN PROGRAMS \ %1.PAS
%1.BAK
REN PROGRAMS \ %1.UPC
%1.PAS
:ENDE
```

Diese Batch-Datei darf nur mit dem Namen des Programms aufgerufen werden, die Dateikennung .PAS darf nicht mit angegeben werden.

Programminternes

Das Hauptprogramm

Im Hauptprogramm wird zuerst die globale Boolesche Variable "abbruch" auf FALSE gesetzt. "abbruch" wird im Programm benutzt, um nach einem Ein-/Ausgabefehler die Verarbeitung abzubrechen. Nach der Ausgabe der Startmeldung des Programms wird die Prozedur Testen aufgerufen, in der sichergestellt wird, daß nur dann eine existierende Datei umgewandelt wird, wenn die Schlüsselwortdatei UPCASE.DAT vorhanden ist. Andernfalls wird "abbruch" auf TRUE gesetzt, und

die folgenden Arbeitsschritte (Einlesen der Schlüsselwortdatei, Quelldatei umwandeln) werden nicht ausgeführt.

Da die Schlüsselwortliste auf dem Heap abgelegt wird, wird der Beginn des Heaps vor dem Einlesen der Schlüsselwörter mit dem Befehl MARK in die Variable "HeapZeiger" gespeichert. Nach dem Abschluß der Umwandlungsarbeiten wird der alte Zustand des Heaps mit Hilfe von RELEASE wiederhergestellt.

PROCEDURE Testen

Hier wird untersucht, ob die Schlüsselwortdatei UPCASE.DAT im aktuellen Verzeichnis gefunden werden kann und ob diese zu öffnen ist.

Nur wenn die Schlüsselwortdatei vorhanden ist, wird geprüft, ob beim Aufruf in der Kommandozeile bereits ein Dateiname übergeben wurde. Wenn nicht, wird der Name der umzuwandelnden Datei im Dialog abgefragt. Dabei ist es egal, ob die Standardendung .PAS für den Dateinamen eingegeben wurde. Wenn im eingegebenen Dateinamen kein '.' vorhanden ist, wird der Dateiname um .PAS ergänzt, wenn nicht, wird der Dateiname unverändert übernommen, wichtig für z.B. Includedateien, die mit der Dateiendung .INC abgespeichert werden.

Anschließend wird versucht, die angegebene Datei zu finden und zu öffnen. Solange dieses nicht gelingt, wird eine erneute Eingabe des Dateinamens angefordert. Wenn es gelingt, wird untersucht, ob die umgewandelte Datei überhaupt noch auf den angegebenen Datenträger paßt. Falls der freie Platz zu gering ist, wird ausgegeben, wieviel Platz auf dem Datenträger mindestens frei sein muß, damit die Datei umgewandelt werden kann, danach wird "abbruch" auf TRUE gesetzt, und damit wird UPCASE abgebrochen.

PROCEDURE Upcase_dat_einlesen

In dieser Prozedur werden die in der Datei UPCASE.DAT vorhandenen Vorgaben der Schlüsselwortschreibweisen in das ARRAY "befehle" eingelesen. "befehle" besteht aus den Startzeigern auf einfach verketteten Listen, wobei sich die einem eingelesenen Schlüsselwort entsprechende Liste aus dem großgeschriebenen Anfangsbuchstaben dieses Schlüsselwortes ergibt.

Jedes Listenelement enthält eine Schlüsselwortvorgabe aus der Datei UPCASE.DAT, das jeweilige Schlüsselwort nur aus Großbuchstaben bestehend zum schnelleren Vergleichen mit den aus dem Text isolierten Bezeichnern und einen Zeiger auf das nachfolgende Listenelement. Bei dem jeweils letzten Element einer Liste hat dieser Zeiger den Wert NIL, was als Endemarkierung benutzt wird.

Zu Beginn werden die ARRAYs "befehle" und "laufzeiger" mit NIL initialisiert. Bei "befehle" bedeutet das, daß keine Schlüsselwörter mit dem entsprechenden Anfangsbuchstaben vorhanden sind, bei "laufzeiger", daß es noch keine Liste für diesen Anfangsbuchstaben gibt.

Nach dem Öffnen der Schlüsselwortdatei UPCASE.DAT werden solange Schlüsselwörter eingelesen, bis das Dateiende erreicht ist. Wenn der Anfangsbuchstabe des eingelesenen Schlüsselwortes in den Indexbereich des ARRAYs "befehle" paßt, wird mit NEW ein Speicherbereich für einen Schlüsselworttyp auf dem Heap allokiert. In diesen Speicherbereich wird das eingelesene Schlüsselwort, das Schlüsselwort in Großbuchstaben und ein mit NIL initialisierter Zeiger eingetragen. Der weitere Ablauf hängt davon ab, ob bereits ein Schlüsselwort mit dem gleichen Anfangsbuchstaben in der Liste eingetragen ist oder nicht. Wenn kein Schlüsselwort eingetragen ist, werden die für diesen Anfangsbuchstaben zuständigen Startzeiger von "befehle" und der Laufzeiger auf den neu geschaffenen Speicherbereich auf dem Heap gesetzt, wobei der Startzeiger auf den ersten Eintrag und Laufzeiger auf den letzten Eintrag der Liste zeigt, in diesem Fall noch auf das gleiche Schlüsselwort.

PROCEDURE Umwandeln

Diese Prozedur erledigt die eigentliche Arbeit des Programms, indem zuerst die umzuwandelnde Quelldatei und die neu zu schaffende Zieldatei geöffnet werden und anschließend die Quelldatei in Blöcken zu jeweils höchstens "maxzeilenanzahl" Zeilen eingelesen, umgewandelt und dann die umgewandelten Zeilen wieder in Blöcken der eingelesen Größe in die Zieldatei geschrieben werden.

Während der Bearbeitung wird die aktuelle Zeilennummer auf dem Bildschirm ausgegeben, damit der Benutzer auch etwas vom Fortgang des Programms zu sehen bekommt.

Die aktuelle Zeile wird Zeichen für Zeichen bearbeitet, wobei zwei Situationen zu unterscheiden sind:

– Die aktuelle Position in der aktuellen Zeile befindet sich nicht in einem Kommentar: Dann muß nach dem Anführungszeichen gesucht werden, um den Inhalt von Zeichenketten aus dem Umwandlungsprozeß herauszuhalten, wobei dann "textbereich" innerhalb der Zeichenketten auf TRUE und außerhalb von Zeichenketten auf FALSE gesetzt wird.

– Die aktuelle Position in der aktuellen Zeile befindet sich nicht in einer Zeichenkette: Dann muß nach dem Beginn bzw. Ende eines Kommentars gesucht werden, wobei auf beide in Pascal möglichen Kommentarkennzeichen zu achten ist: '{' und '(*' als Kommentarbeginn und '}' und '*)' als Kommentarende.

Innerhalb eines mit '{...}' gebildeten Kommentars ist "kommentar1" TRUE, innerhalb eines mit '(*...*)' gebildeten Kommentars ist "kommentar2" TRUE.

Beide Kommentare können zwar ineinander verschachtelt werden, es muß aber nur auf den ersten vorkommenden Kommentarbeginn geachtet werden, da das Kommentarende zum Kommentaranfang passen muß.

Wenn sich die aktuelle Position in einem Kommentar oder in einem Textbereich befindet, wird die aktuelle Position um ein Zeichen erhöht, sonst wird entschieden, ob "Wort_isolieren" aufgerufen wird. Diese Entscheidung wird entsprechend der Pascal-Notation für Bezeichner und Variablen getroffen, das heißt, das erste Zeichen muß ein Buchstabe sein, auf den weitere Buchstaben, Ziffern oder der Unterstrich '_' folgen können. Nur wenn an der aktuellen Position ein Buchstabe gefunden wird, wird in die Funktion "Wort_isolieren" verzweigt, wobei die aktuelle Zeile und die aktuelle Position in dieser Zeile übergeben werden.

FUNCTION Wort_isolieren

Diese Funktion isoliert in der aktuellen Zeile ein Wort, indem das Wort aus dem Buchstaben an der übergebenen Position und den folgenden Zeichen aufgebaut wird, solange diese weiteren Zeichen aus Buchstaben, Ziffern und dem Unterstrich '_' bestehen.

Danach wird das isolierte Wort in Großbuchstaben umgewandelt und mit den Inhalten der Liste seines Anfangsbuchstaben verglichen. Dazu wird "laufzeiger" auf den ersten Eintrag der Liste gesetzt und solange auf das nächste Listenelement weitergeschaltet, bis entweder das Ende der Liste erreicht ist (Wort nicht gefunden -> isoliertes Wort ist kein Schlüsselwort) oder das isolierte Wort mit dem Vergleichswort des Listenelements übereinstimmt. In diesem Fall wird das isolierte Wort in der Zeile durch die in dem Listenelement vorhandene Vorgabe ersetzt.



PROCEDURE IOFehler
Diese Prozedur wird nach jeder Ein-/Ausgabeoperation aufgerufen. Hier wird der Wert von "IOResult" ausgewertet. Falls dieser ungleich Null ist, ist ein Fehler aufgetreten, und die zur Fehlernummer gehörende Fehlermeldung wird ausgegeben. Außerdem wird die globale Variable "abbruch" auf TRUE gesetzt und damit die weitere Bearbeitung der Datei unterbrochen.

Abtipphilfe

Zuerst tippen Sie die drei Listings "FILEINOU.PAS", "INOUT.PAS" und "UPCASE.PAS" mittels Turbo-Pascal-Compiler oder beliebiger Textverarbeitung (ASCII-Format) ab. Die Schlüsselwortdatei "UPCASE.DAT" ist ebenfalls im ASCII-Format auf den Datenträger zu bringen. Aus Platzgründen haben wir die Worte dieser Liste hintereinandergeschrieben. Beim Abtippen gönnen Sie bitte jedem Begriff eine neue Zeile. Natürlich ist diese Datei beliebig erweiter- bzw. veränderbar, sofern man sich an das vorgegebene Format hält. Dadurch ist es durchaus denkbar, auch die in anderen Programmiersprachen erstellten Programme durch UPCASE in ein leserliches Format zu bringen.
Nach dem Abtippen muß das Listing UPCASE.PAS mittels Pascal-Compiler 5.0 compiliert werden. Auf der DATABOX ist das bereits lauffähige Programm sowie deren Quellcode vorhanden. (Dirk Alpers/jf)

Abs	*ABSOLUTE*	*Addr*	*AND*	*Append*	*Arc*
ArcTan	*ARRAY*	*Assign*	*AssignCrt*	*Bar*	*Bar3D*
BEGIN	*BlockRead*	*BlockWrite*	*BOOLEAN*	*BYTE*	*CASE*
CHAR	*ChDir*	*Chr*	*Circle*	*ClearDevice*	*ClearViewPort*
Close	*CloseGraph*	*ClrEol*	*ClrScr*	*Concat*	*CONST*
Copy	*Cos*	*Crt*	*CSeg*	*Dec*	*Delay*
Delete	*DelLine*	*DetectGraph*	*DiskFree*	*DiskSize*	*Dispose*
DIV	*DO*	*Dos*	*DosError*	*DosExitCode*	*DosVersion*
DOWNTO	*DrawPoly*	*DSeg*	*Ellipse*	*ELSE*	*END*
EnvCount	*EnvStr*	*EoF*	*EoLn*	*Erase*	*Exec*
Exit	*Exp*	*EXTERNAL*	*FALSE*	*FExpand*	*FILE*
FilePos	*FileSize*	*FillChar*	*FillEllipse*	*FillPoly*	*FindFirst*
FindNext	*FloodFill*	*Flush*	*FOR*	*FORWARD*	*Frac*
FreeMem	*FSearch*	*FSplit*	*FUNCTION*	*GetArcCoords*	*GetAspectRatio*
GetBkColor	*GetCBreak*	*GetColor*	*GetDate*	*GetDefaultPalette*	*GetDir*
GetDriverName	*GetEnv*	*GetFAttr*	*GetFillPattern*	*GetFillSettings*	*GetFTime*
GetGraphMode	*GetImage*	*GetIntVec*	*GetLineSettings*	*GetMaxColor*	*GetMaxMode*
GetMaxX	*GetMaxY*	*GetMem*	*GetModeName*	*GetModeRange*	*GetPalette*
GetPaletteSize	*GetPixel*	*GetTextSettings*	*GetTime*	*GetVerify*	*GetViewSettings*
GetX	*GetY*	*GOTO*	*GotoXY*	*Graph*	*GraphDefaults*
GraphErrorMsg	*GraphResult*	*Halt*	*Hi*	*HighVideo*	*IF*
ImageSize	*IMPLEMENTATION*	*IN*	*Inc*	*InitGraph*	*INLINE*
Insert	*InsLine*	*InstallUserDrive*	*InstallUserFont*	*Int*	*INTEGER*
INTERFACE	*INTERRUPT*	*Intr*	*Invers*	*IOResult*	*Keep*
KeyPressed	*LABEL*	*Length*	*Line*	*LineRel*	*LineTo*
Ln	*Lo*	*LONGINT*	*LowVideo*	*LST*	*Mark*
MaxAvail	*MemAvail*	*MkDir*	*MOD*	*Move*	*MoveRel*
MoveTo	*MsDos*	*New*	*NIL*	*Normal*	*NormVideo*
NoSound	*NOT*	*Odd*	*OF*	*Ofs*	*OR*
Ord	*OutText*	*OutTextXY*	*Overlay*	*OvrClearBuf*	*OvrGetBuf*
OvrInit	*OvrInitEMS*	*OvrSetBuf*	*PACKED*	*PackTime*	*ParamCount*
ParamStr	*Pi*	*PieSlice*	*Pos*	*Pred*	*Printer*
PROCEDURE	*PROGRAM*	*Ptr*	*PutImage*	*PutPixel*	*Random*
Randomize	*Read*	*ReadKey*	*ReadLn*	*REAL*	*RECORD*
Rectangle	*RegisterBGIDriver*	*RegisterBGIFont*	*Release*	*Rename*	*REPEAT*
Reset	*RestoreCrt*	*RestoreCrtMode*	*Rewrite*	*RmDir*	*Round*
RunError	*SearchRec*	*Sector*	*Seek*	*SeekEoF*	*SeekEoLn*
Seg	*SET*	*SetActivePage*	*SetAllPalette*	*SetAspectRatio*	*SetBkColor*
SetCBreak	*SetColor*	*SetDate*	*SetFAttr*	*SetFillPattern*	*SetFillStyle*
SetFTime	*SetGraphBufSize*	*SetGraphMode*	*SetIntVec*	*SetLineStyle*	*SetPalette*
SetRGBPalette	*SetTextBuf*	*SetTextJustify*	*SetTextStyle*	*SetTime*	*SetUserCharSize*
SetVerify	*SetViewPort*	*SetVisualPage*	*SetWriteMode*	*SHL*	*SHORTINT*
SHR	*Sin*	*SizeOf*	*Sound*	*SPtr*	*Sqr*
Sqrt	*SSeg*	*Str*	*StrExpand*	*STRING*	*Succ*
Swap	*SwapVectors*	*TEXT*	*TextBackground*	*TextColor*	*TextHeight*
TextMode	*TextWidth*	*THEN*	*TO*	*TRUE*	*Trunc*
Truncate	*TYPE*	*UNIT*	*UnpackTime*	*UNTIL*	*UpCase*
USES	*Val*	*VAR*	*Warten*	*WhereX*	*WhereY*
WHILE	*Window*	*WITH*	*WORD*	*Write*	*WriteLn*
WritelnTab	*WriteTab*	*XOR*			

```
UNIT InOut;
{ Proceduren zur Ein- und Ausgabe im Textmodus

  Dirk Alpers
  Feldstraße 135  App. 100
  2000 Wedel / Holstein
}

INTERFACE

FUNCTION StrUpcase ( lstr: STRING ): STRING;
{ Upcase für den gesamten String mit Umwandlung der Umlaute ä, ü, ö }

{===========================================================================}

IMPLEMENTATION

FUNCTION StrUpcase ( lstr: STRING ): STRING;
{ Upcase für den gesamten String mit Umwandlung der Umlaute ä, ü, ö

  Parameter:  lstr : die in Großbuchstaben umzuwandelnde Zeichenkette

  Ergebnis :  die in Großbuchstaben umgewandelte Zeichenkette

  Dirk Alpers
}
VAR  lix : BYTE;                              { Schleifenzähler                }
     ch  : CHAR;                              { einzelnes Zeichen der Zeichenkette }
BEGIN
  FOR lix := 1 TO Length(lstr) DO BEGIN
    ch := lstr[lix];
    IF (ch >= 'a') AND (ch <= 'z')
      THEN lstr[lix] := UpCase(ch)
      ELSE CASE ch OF
             'ä': lstr[lix] := 'Ä';
             'ö': lstr[lix] := 'Ö';
             'ü': lstr[lix] := 'Ü';
           END; { case }
    END; { for }

  StrUpcase := lstr;
END; { StrUpcase }

{===========================================================================}

END. { InOut }
```

Listing UPCASE

```
UNIT FileInOut;
{ Hilfsroutinen zur Dateibehandlung

  Dirk Alpers
  Feldstraße 135  App. 100
  2000 Wedel / Holstein
}

INTERFACE

FUNCTION FileExist ( lstr: STRING ): BOOLEAN;
{ überprüft, ob sich die angegebene Datei öffnen läßt }

FUNCTION IOResultMsg ( fehlercode: BYTE ): STRING;
{ Fehlermeldungstext zu einem aufgetretenen Laufzeitfehler }

{===========================================================================}

IMPLEMENTATION

FUNCTION FileExist ( lstr: STRING ): BOOLEAN;
{ Überprüft, ob sich die angegebene Datei öffnen läßt

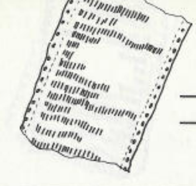
  Parameter:  lstr  : Pfad und/oder Dateiname in DOS-Syntax

  Ergebnis :  TRUE  : Datei gefunden und nicht schreibgeschützt
              FALSE : Datei nicht gefunden oder schreibgeschützt

  Dirk Alpers
}
VAR  datei    : FILE OF BYTE;             { zu untersuchende Datei }
     iofehler : INTEGER;                  { Ergebnis von IOResult  }
BEGIN
  iofehler := IOResult;
  {$I-}
  Assign(datei, lstr);
  Reset(datei);
  IF (IOResult = 0)
    THEN FileExist := TRUE
    ELSE FileExist := FALSE;
  Close(datei);
  {$I+}
  iofehler := IOResult;
END; { FileExist() }

{---------------------------------------------------------------------------}
```

Listing UPCASE



```
FUNCTION IOResultMsg ( fehlercode: BYTE ): STRING;
{ Fehlermeldungstext zu einem aufgetretenen Laufzeitfehler bei der Ein- und
  Ausgabe

  Parameter:  fehlercode : Nummer des aufgetretenen Fehlers

  Ergebnis :  Text der Fehlermeldung zu dem aufgetretenen Fehler

  Dirk Alpers
}
VAR  lstr : STRING;                        { Fehlermeldungstext }
BEGIN
  CASE fehlercode OF
    { Fehlercodes von DOS }
      2: lstr := 'Datei nicht gefunden';
      3: lstr := 'Pfad nicht gefunden';
      4: lstr := 'Maximalanzahl an Dateien bereits offen';
      5: lstr := 'Dateizugriff verweigert';
      6: lstr := 'Handle nicht definiert/ungültig';
     12: lstr := 'Ungültiger Dateimodus';
     15: lstr := 'Laufwerksmodus unzulässig';
     16: lstr := 'Das als Standard gesetzte DIR kann nicht gelöscht werden';
     17: lstr := 'RENAME kann nicht kopieren';

    { Fehler bei der Bearbeitung von Dateien }
    100: lstr := 'Fehler beim Lesen von Diskette';
    101: lstr := 'Fehler beim Schreiben auf Diskette';
    102: lstr := 'Datei-Variable ist keiner Datei zugeordnet';
    103: lstr := 'Datei ist nicht offen';
    104: lstr := 'Datei ist nicht zum Lesen geöffnet';
    105: lstr := 'Datei ist nicht zum Schreiben geöffnet';
    106: lstr := 'Ungültiges numerisches Format';

    { Kritische Fehler }
    150: lstr := 'Diskette ist schreibgeschützt';
    151: lstr := 'Peripheriegerät nicht bekannt/nicht angeschlossen';
    152: lstr := 'Laufwerk nicht betriebsbereit';
    153: lstr := 'Ungültiger DOS-Funktionscode/Funktion nicht definiert';
    154: lstr := 'Prüfsummenfehler beim Lesen von Diskette/Festplatte';
    155: lstr := 'Ungültiger Disk-Parameterblock';
    156: lstr := 'Kopf-Positionierungsfehler';
    157: lstr := 'Unbekanntes Sektorformat';
    158: lstr := 'Diskettensektor nicht lokalisierbar';
    159: lstr := 'Kein Papier im Drucker';
    160: lstr := 'Schreibfehler beim Zugriff auf ein Peripheriegerät';
    161: lstr := 'Lesefehler beim Zugriff auf ein Peripheriegerät';
    162: lstr := 'Nicht genauer bestimmbarer Hardware-Fehler';
    ELSE lstr := '';
    END; { case }

  IOResultMsg := lstr;
END; { IOResutMsg() }

{==========================================================================}

END. { FileInOut }
```

```
{$A+,B-,D+,E+,F-,I+,L+,N+,O-,R+,S+,V+}
{$M 65520,20000,635550}
PROGRAM UpCase;

{ wandelt die in einem Turbo Pascal-Programm vorkommenden Schlüsselwörter
  mit Hilfe der in "UPCASE.DAT" enthaltenen Vorgaben in großgeschriebene um
  SW=Schlüsselwort
  Dirk Alpers
}

USES  InOut,                          { StrUpcase                }
      FileInOut,                      { FileExist, IOResultMsg }
      Crt,
      Dos;

CONST  befehlslaenge    = 40;           {maximale Länge eines Befehls}
       upcase_dat       = 'UPCASE.DAT';{Name der Schlüsselwortdatei}
       ersterbuchstabe  = 'A';          {1. möglicher Buchstabe eines SW}
       letzterbuchstabe = 'Z';          {letzter möglicher Buchstabe eines SW}
       umzuwandeln      = '.PAS';       {Dateikennung der umzuwandelnden Datei}
       umgewandelt      = '.UPC';       {Dateikennung der umgewandelten Datei}

TYPE   zeiger            = ^schluesselworttyp;
       schluesselworttyp = RECORD
                             vorgabe    : STRING[befehlslaenge];
                             vergleich  : STRING[befehlslaenge];
                             nachfolger : zeiger;
                             END; { record }

VAR    abbruch     : BOOLEAN;
       befehle     : ARRAY [ersterbuchstabe..letzterbuchstabe]
                        OF zeiger;
       dateiname   : STRING;
       HeapZeiger  : Pointer;

{==========================================================================}

PROCEDURE IOFehler ( fehlernummer: INTEGER );
{ gibt bei einem eventuell aufgetretenen Ein-/Ausgabefehler die dazugehörige
  Fehlermeldung aus und setzt die globale Variable abbruch auf TRUE

  Parameter:  fehlernummer : Ergebnis von IOResult()

  Dirk Alpers
}
BEGIN
  IF (fehlernummer <> 0)
    THEN BEGIN
         WriteLn(#7, #7, IOResultMsg(fehlernummer));
         abbruch := TRUE;
         END;
END; { IOFehler }

{--------------------------------------------------------------------------}

PROCEDURE testen;
{ Test auf:    1) Schlüsselwortdatei im aktuellen Verzeichnis vorhanden ?
               2) Noch genügend Platz für die umgewandelte Datei auf dem
                  Datenträger vorhanden ?

  Dirk Alpers
}
VAR  dateiinfo : SearchRec;          { Größe der umzuwandelnde Datei ermitteln }
BEGIN

  IF NOT FileExist(upcase_dat){ UPCASE.DAT im aktuellen Verzeichnis vorhanden ?}
    THEN BEGIN
         WriteLn(#7, #7, 'Schlüsselwortdatei ', upcase_dat, 'nicht gefunden !');
         abbruch := TRUE;
         END;

  IF NOT abbruch
    THEN BEGIN
         IF (ParamCount > 0)             { Dateinamen beim Aufruf übergeben ?  }
           THEN BEGIN                    { ja, aus der Kommandozeile holen     }
                dateiname := ParamStr(1);
                WriteLn;
                END
           ELSE BEGIN                    { nein, Dateinamen im Dialog einlesen }
                Write('Umzuwandelnde Datei: ');
                ReadLn(dateiname);
                WriteLn;
                WriteLn;
                END;
```

Listing UPCASE

```
         IF (Pos('.', dateiname) = 0)          { eventuell '.PAS' ergänzen }
           THEN dateiname := dateiname + umzuwandeln;

         WHILE NOT FileExist(dateiname) DO BEGIN        { Dateinamen einlesenn }
           WriteLn(FExpand(dateiname), ' nicht gefunden.  Bitte neu eingeben.');
           Write('Umzuwandelnde Datei: ');
           ReadLn(dateiname);
           WriteLn;
           WriteLn;
           IF (Pos('.', dateiname) = 0)
             THEN dateiname := dateiname + umzuwandeln;
           END; { while }

         FindFirst(dateiname, anyfile, dateiinfo); { Dateigröße ermitteln }
         IF (dateiinfo.size > DiskFree(0)) { zu wenig Platz auf dem Datenträger}
           THEN BEGIN
                WriteLn('Nicht mehr genügend Speicherpl. auf der Disk frei.');
                WriteLn('Es fehlen ', dateiinfo.size - DiskFree(0), ' Bytes.');
                abbruch := TRUE;
                END;

         END; { if not abbruch }

END; { Testen }

{--------------------------------------------------------------------------}

PROCEDURE Upcase_dat_einlesen;
{ Vorgaben der Schlüsselwörter je nach Anfangsbuchstabe in die jeweilige
  einfach verkettete Liste einlesen

  befehle besteht aus den Startzeigern auf einfach verketteten Listen, wobei
  sich die einem eingelesenen Schlüsselwort entsprechende Liste aus dem
  großgeschriebenen Anfangsbuchstaben des Schlüsselwortes ergibt
}
VAR  anfangsbuchstabe : CHAR;            { 1. Buchstabe des Schlüsselworts    }
     befehlswort      : STRING[befehlslaenge]; {Schlüsselwort aus der Datei }
     datei            : TEXT;              { Datei der Schlüsselwörter        }
     laufzeiger       : ARRAY [ersterbuchstabe..letzterbuchstabe]
                           OF zeiger;
     lcx              : CHAR;            { Schleifenzähler                    }
     tempzeiger       : zeiger;          { Temporärer Schlüsselworteintrag    }
BEGIN
  { Schlüsselwortliste und Zeiger auf Listenende initialisieren }
  FOR lcx := ersterbuchstabe TO letzterbuchstabe DO BEGIN
    befehle[lcx] := NIL;
    laufzeiger[lcx] := NIL;
    END;

  {$I-}
  Assign(datei, upcase_dat);
  Reset(datei);                          { Schlüsselwortdatei zum Lesen öffnen }
  iofehler(IOResult);

  WHILE NOT EoF(datei) AND NOT abbruch DO BEGIN  { Befehle in Liste einlesen }
    ReadLn(datei, befehlswort);
    iofehler(IOResult);

    anfangsbuchstabe := befehlswort[1];

    { die dem Anfangsbuchstaben des Schlüsselworts entsprechende Liste }
    { anlegen bzw. erweitern                                           }
IF (ersterbuchstabe<=anfangsbuchstabe) AND (anfangsbuchstabe<=letzterbuchstabe)
      THEN BEGIN
           New(tempzeiger);              { neuen Listeneintrag vorbereiten }
           tempzeiger^.vorgabe := befehlswort;
           tempzeiger^.vergleich := StrUpcase(befehlswort);
           tempzeiger^.nachfolger := NIL;

           IF (befehle[anfangsbuchstabe] = NIL)
             THEN BEGIN                  { erster Eintrag in einer Liste }
                  befehle[anfangsbuchstabe] := tempzeiger;
                  laufzeiger[anfangsbuchstabe] := tempzeiger;
                  END
             ELSE BEGIN                  { Nachfolgeeintrag in eine Liste }
                  laufzeiger[anfangsbuchstabe]^.nachfolger := tempzeiger;
                  laufzeiger[anfangsbuchstabe] := tempzeiger;
                  END;

           END; { ersterbuchstabe <= anfangsbuchstabe <= letzterbuchstabe }

    END; { while not eof(datei) and not abbruch }

  Close(datei);                          { Schlüsselwortdatei schliessen }
  iofehler(IOResult);
  {$I+}
END; { Upcase_dat_einlesen }

{--------------------------------------------------------------------------}

PROCEDURE Umwandeln;
{ durch die gesamte umzuwandelnde Datei gehen und alle Schlüsselwörter in
  in ihre Vorgaben umwandeln

  Dirk Alpers
}
CONST  buchstaben      = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
       zahlen          = '0123456789_';
       maxzeilenanzahl = 100; { Höchstanzahl der gleichzeitig gelesenen Zeilen }
VAR    quelldatei     : TEXT;  { umzuwandelnde Datei                    }
       zieldatei      : TEXT;  { umgewandelte Datei                     }
       kommentar1     : BOOLEAN; { Kommentar in geschweiften Klammern    }
       kommentar2     : BOOLEAN; { Kommentar in Klammern und Sternen     }
       textbereich    : BOOLEAN; { Text in Anführungszeichen            }
       lix            : INTEGER; { Schleifenzähler                       }
       zeilenzaehler  : WORD;    { Anzahl der verarbeiteten Zeilen       }
       zeile          : ARRAY [1..maxzeilenanzahl]   { eingelesene Zeilen }
                           OF STRING;
       zeilenanzahl   : BYTE;      { Anzahl der eingelesenen Zeilen       }
       ausgabezeile   : BYTE;      { Bildschirmzeile, in der die aktuelle }
                                   {   Zeilennummer ausgegeben wird       }
       ch             : CHAR;      { Zeichen innerhalb der aktuellen Zeile }
       position       : BYTE;      { Position innerhalb der aktuellen Zeile }

  {--------------------------------}

  FUNCTION wort_isolieren ( VAR zeile: STRING; position: BYTE ): BYTE;
  { in der aktuellen Zeile ein einzelnes Wort isolieren, das mit einem
    Buchstaben beginnt und aus folgenden Buchstaben, Ziffern oder '_'
    zusammengesetzt ist

    Parameter:  zeile    : aktuelle Zeile
                position : aktuelle Untersuchungsposition in der aktuellen
                           Zeile

    Ergebnis :  als Funktionsergebnis: die Position hinter dem isolierten Wort
                als VAR-Parameter: die Zeile mit dem eventuell jetzt großge-
                  schriebenen isolierten Wort, falls es ein Schlüsselwort war

    Dirk Alpers
  }
  VAR  wort        : STRING[befehlslaenge];{isoliertes Wort                 }
       lix         : BYTE;                 {Schleifenzähler                  }
       laufzeiger  : zeiger;               {Zeiger, der eine Liste abläuft, bis }
                                           {  das Schlüsselwort gefunden ist    }
                                           {  oder das Listenende erreicht ist  }
  BEGIN
    wort := '';

 {Wort aus der Zeile isolieren,solange es aus Buchstaben,Ziffern & '_' besteht }
    lix := position;
  WHILE (lix <= Length(zeile)) AND ((Pos(Copy(zeile, lix, 1), buchstaben) > 0) OR
      (Pos(zeile[lix], zahlen) > 0)) DO BEGIN
      wort := wort + zeile[lix];
      Inc(lix);
      END; { while }
```

Listing UPCASE




```
    wort := StrUpcase(wort);

    { durch die jeweilige Liste laufen, bis das Wort gefunden ist oder das }
    { Listenende erreicht ist                                               }
    laufzeiger := befehle[wort[1]];
    WHILE (laufzeiger^.nachfolger <> NIL) AND (laufzeiger^.vergleich <> wort) DO
      laufzeiger := laufzeiger^.nachfolger;

    { wenn das Wort in der Liste vorhanden ist, in der Zeile das Wort nach }
    { der Vorgabe gemischt oder nur großschreiben                           }
    IF (laufzeiger^.vergleich = wort)
      THEN FOR lix := 1 TO Length(wort) DO
             zeile[position + lix - 1] := laufzeiger^.vorgabe[lix];

    wort_isolieren := position + Length(wort);
  END; { wort_isolieren }

  {----------------------------------}

BEGIN { Umwandeln }
  kommentar1 := FALSE;                   { nicht im Kommentarbereich   }
  kommentar2 := FALSE;
  textbereich := FALSE;                  { nicht in einem Textbereich  }
  zeilenzaehler := 0;                    { noch keine Zeile bearbeitet }
  ausgabezeile := WhereY - 1;            { Zeile auf dem Bildschirm    }

  {$I-}
  { angegebene Datei zum Lesen öffnen  }
  Assign(quelldatei, dateiname);
  Reset(quelldatei);
  iofehler(IOResult);

  { Dateiname.UPC zum Schreiben öffnen }
  IF NOT abbruch
    THEN BEGIN
   Assign(zieldatei, Copy(dateiname, 1, Pos('.', dateiname) - 1) + umgewandelt);
         Rewrite(zieldatei);
         iofehler(IOResult);
         END;

  { gesamte Quelldatei einlesen }
  WHILE NOT EoF(quelldatei) AND NOT abbruch DO BEGIN
    zeilenanzahl := 0;

    { maxzeilenanzahl Zeilen aus der Quelldatei einlesen }
WHILE NOT EoF(quelldatei) AND (zeilenanzahl<maxzeilenanzahl) AND NOT abbruch DO
      BEGIN
      Inc(zeilenanzahl);
      ReadLn(quelldatei, zeile[zeilenanzahl]);
      iofehler(IOResult);
      END; { while not eof(quelldatei) and ... }

    { eingelesene Zeilen umwandeln }
    FOR lix := 1 TO zeilenanzahl DO BEGIN
      Inc(zeilenzaehler);                { aktuelle Gesamtzeilenanzahl auf dem }
      GotoXY(1,ausgabezeile);            { Bildschirm ausgeben                 }
      Write(zeilenzaehler);

      position := 1;                     { 1. Buchstabe einer Zeile }
      WHILE (position <= Length(zeile[lix])) DO BEGIN
        ch := zeile[lix][position];

        IF NOT (kommentar1 OR kommentar2)   { Zeichenkettenbegrenzungen }
          THEN IF (ch = '''')
                 THEN textbereich := NOT textbereich;

        IF NOT textbereich                  { Kommentarbegrenzungen }
          THEN BEGIN
               IF NOT kommentar2
                 THEN BEGIN
                      IF (ch = '{')
                        THEN kommentar1 := TRUE;
                      IF (ch = '}')
                        THEN kommentar1 := FALSE;
                      END;

               IF NOT kommentar1
                 THEN BEGIN
                      IF (Copy(zeile[lix], position, 2) = '(*')
                        THEN BEGIN
                             kommentar2 := TRUE;
                             Inc(position);
                             END;
                      IF (Copy(zeile[lix], position, 2) = '*)')
                        THEN BEGIN
                             kommentar2 := FALSE;
                             Inc(position);
                             END;
                      END;
               END;

        IF kommentar1 OR kommentar2 OR textbereich
          THEN Inc(position)      { Kommentare oder Textbereiche nicht umwandeln }
          ELSE IF (Pos(ch, buchstaben) > 0)
                 THEN position := wort_isolieren(zeile[lix], position)
                 ELSE Inc(position);

        END; { while (position < length(zeile)) }
      END; { for }

    IF NOT abbruch
      THEN FOR lix := 1 TO zeilenanzahl DO BEGIN {umgewandelte Zeilen schreiben}
             WriteLn(zieldatei, zeile[lix]);
             iofehler(IOResult);
             END; { for }

    END; { while not eof(quelldatei) and not abbruch }

  Close(quelldatei);                     { angegebene Datei schließen }
  iofehler(IOResult);
  Close(zieldatei);                      { Dateiname.UPC schließen    }
  iofehler(IOResult);
  {$I+}
END; { Umwandeln }

{============================ Hauptprogramm ================================}

BEGIN
  abbruch := FALSE;

  WriteLn;
  WriteLn('UPCASE für Turbo Pascal');
  WriteLn;

  Testen;

  Mark(HeapZeiger);                      { freien Bereich des Heaps merken }

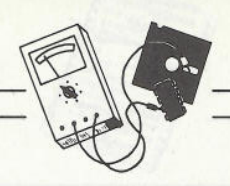
  IF NOT abbruch
    THEN upcase_dat_einlesen;            { Befehlstabelle einlesen          }
  IF NOT abbruch
    THEN umwandeln;                      { angegebene Datei bearbeiten      }
  Release(HeapZeiger);                   { Heap wieder freigeben            }

  IF abbruch
    THEN BEGIN
         WriteLn;
         WriteLn(#7, #7, 'Abbruch des Programms wegen Schreib-/Lesefehler');
         Halt(1);
         END
    ELSE BEGIN
         WriteLn;
         Write(StrUpcase(dateiname), ' in ', StrUpcase(Copy(dateiname, 1,
           Pos('.', dateiname) - 1)), umgewandelt, ' umgewandelt');
         END;

END. { Upcase }
```

Listing UPCASE





The Quest goes on...

Sierras Neue

Aus dem Kino sind wir es gewohnt: Rocky IV, Police Academy V, Das Übliche VI, Und noch 'ne Folge VII...
Was einmal den Hauch des Erfolges genossen hat – also dem Produzenten einen vollen Geldbeutel bescherte – wird gnadenlos in immer neuen, meist immer schlechteren Folgen dem Publikum vorgeworfen.
Sierra droht langsam aber sicher dieses Kino-Syndrom auch auf der Ebene der Computerspiele aufzugreifen. Ob sich dieser negative Beigeschmack des Serienbooms auch in der Qualität der Spiele niederschlägt, haben wir genauer untersucht.



Heros Quest ist Sierras erster Versuch, das bewährte Spielkonzept mit Elementen des Rollenspiels zu verknüpfen. Wahlweise übernehmen Sie die Rolle eines Kämpfers, Diebs oder die eines Magiers. Diese Figuren sind mit Eigenschaften auszustatten, die im Laufe des Spiels verbessert werden können. Dann geht es aber hinaus in eine Welt voller Abenteuer und Mysterien.
Alles beginnt in der kleinen Stadt Spielburg, in der Sie als unerfahrener Gefahrensucher die Irrfahrt beginnen. In der Gilde der Abenteurer findet sich dann auch schnell ein alter müder Krieger, der so manchen Kampf mit Drachen und anderen Getümen hinter sich hat. Von ihm und von einer vollbehängten Pinwand erfährt man, mit welchen Heldentaten man seinen geschrumpften Geldbeutel aufbessern kann.

Erschlagender Umfang

Man steuert seinen Pixelhelden durch Gebäude, Wälder und finstere Gassen, lernt verschiedene Kunstfertigkeiten beherrschen und ist schon bald ein gefürchteter Abenteurer.

Weniger umwerfend, dafür mehr erschlagend wirkt der Umfang, in dem Hero's Quest geliefert wird. Zehn Disketten deuten die Ernsthaftigkeit der Empfehlung, eine Festplatte zu besitzen, um dieses Programm komfortabel laufen zu lassen, in aller Deutlichkeit an. Ebenfalls empfohlen wird ein Prozessor, der mit mindestens 8 MHz getaktet ist. Auch dieser Punkt sollte vom Spieler ernst genommen werden.

Diesen Hardware-Anforderungen (es geht auch ohne, nur dauert es dann entsprechend lange und Ihre Experience-Points als Disk(etten)-Jockey steigen in nicht unerheblichem Maße) stehen Hardware-Unterstützungen entgegen, die sich sehen lassen. Jede nur denkbare Soundkarte wird von Hero's Quest angesteuert; die musikalische Untermalung ist dann dementsprechend hörenswert.

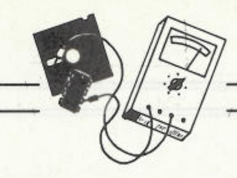
Bei den Grafikkarten ist das Programm ebenfalls nicht wählerisch; ob man im Besitz einer Hercules- oder VGA-Karte ist, es treten keine Probleme auf, wenn sich auch der versprochene VGA-Modus als angepaßte EGA-Grafik entpuppt.

Der Spielverlauf selbst hat sich im Vergleich zu anderen Sierra-Spielen wenig geändert. In bewährter Weise steuert man seinen Helden von Bild zu Bild. Es ist hier in ebenfalls bewährter Weise empfehlenswert, zur Steuerung des Abenteurers die Tastatur zu benutzen, da die Maussteuerung noch immer nicht ausgereift ist. So kann es zu einem Abenteuer im Abenteuer werden, will man den Helden mit der Maus von einem ins andere Bild bewegen.
Die in keinem Rollenspiel fehlenden Kampfsequenzen heben sich in Hero's Quest wohltuend von den üblichen Gemetzeln ab. Hier steht man seinem Gegner gegenüber und kämpft mit verschiedenen Abwehr- und Angriffsaktionen gegen Monster und Unholde. Eine gute Idee, die allerdings in der Ausführung noch einige Schwächen aufweist. Die Animation, der in dieser Phase sehr großen Figuren, ist für ein auch auf Schnelligkeit bedachtes Kampfgeschehen einfach zu träge.
Ein gelungener Einfall ist eine neue Funktion, die es ermöglicht, Gegenstände innerhalb des Bildes direkt anzuklicken, um so an spezielle Informationen zu kommen. Das erspart die oft zeitraubende Suche nach irgendwelchen speziellen Vokabeln, die der Parser verlangt. Stichwort Parser: Daß der Parser in Sierra-Spielen bislang meist der schwächste Programmteil war, dürfte hinreichend bekannt sein. Leider wurden diesbezüglich auch bei Hero's Quest keinerlei Verbesserungen eingeführt. So schafft es dieser Eingabe-Interpreter nicht, mehrere Befehle in Folge zu verarbeiten, was sich mitunter als störend erweist.
Eine eindeutige Aussage über die im Programm zu lösenden Probleme ist schwer zu formulieren. Denjenigen, die es knifflig und logisch ergründbar





Trotz Farb- und Monstervielfalt vermag Hero's Quest nicht zu überzeugen
Screenshot: Hero's Quest



mögen, wird hier sicherlich nicht das Mekka des Rätselratens geboten, dazu sind einige Problemlösungen zu sehr dem Zufall überlassen. Wer allerdings Freude am Probieren hat und nicht mit der Holmesschen Meßlatte an das Programm herantritt, der wird auf seine Kosten kommen.

Verzeihbare Mängel

Ein Wort zum Rollenspielcharakter. Man kann ihn als vorhanden ansehen, was nicht heißen soll, daß in diesem Punkt einige Verbesserungen wohltuend auf das Gesamtkonzept wirken würden. Zwar hängen Erfolge in bestimmten Aktionen von der Höhe der dazugehörigen Eigenschaftswerte ab, doch fehlt die für ein gutes Rollenspiel übergeordnete Komplexität und gegenseitige Beeinflussung diverser Eigenschaften. Ein Punkt, der sicherlich bei den nächsten Folgen dieser Reihe aufgearbeitet wird.

Da es sich bei Hero's Quest um das Erstlingswerk der Rollenspiele von Sierra handelt, kann man die diversen Kinderkrankheiten noch entschuldigen. Der Umfang des Spiels steht zwar in noch keinem sehr positiven Verhältnis zur Qualität, doch kann man erwarten, daß sich die Mängel bei der nächsten "Heldensuche" in Luft aufgelöst haben. Eine nette Unterhaltung bietet Hero's Quest allemal, und wem es gefällt, der kommt sicher auf seine Kosten.

Larry, die dritte

Der mittlerweile dritte Teil von **Leisure Suit Larry** trägt den pompösen Untertitel "Passionate Patti in Pursuit of the Pulsating Pectorals". Ebenso voluminös wie dieser Titel ist der Umfang, in dem das Programm geliefert wird. Er kommt zwar nicht ganz an den der "Heldensuche" heran, doch ist mit acht Disketten auch fast schon die Schmerzgrenze des Erträglichen erreicht.

Die Machart des Spiels ist die gleiche wie bei Hero's Quest oder jedem anderen Sierra-Abenteuerspiel. Der Einfallsreichtum und die Ideenfülle sorgen allerdings für eine gehörige Portion Kurzweil.

Das beginnt (wie sollte es anders sein) am Anfang. Der Abfrage nach der Altersgruppe folgt ein kleiner, aus fünf Fragen bestehender Test, der weniger Auswirkungen auf das Spiel, dafür mehr auf eine am Spielbeginn ablaufende voyeuristische Handlung hat.

Wie in den ersten beiden Larry-Abenteuern ist auch in Teil drei der Schwierigkeitsgrad nicht besonders hoch. Die zu erreichende maximale Punktezahl von sage und schreibe 4000 (in Worten: viertausend!) erweist sich nach den ersten Lösungen als sehr relativ, da es für bestimmte positiv zu Ende gebrachte Aktionen bis zu hundert Punkte auf einen Schlag geben kann.

Letztlich kommt es bei Larry III ja auch weniger auf die besonders schwierigen Rätsel an, sondern auf das besondere Flair des Spiels. Das Programm ist stets für einen lockeren Spruch gut und weiß den Spieler mit vielen intelligenten Anspielungen zu unterhalten.



Nichts für prüde Zeitgenossen

Wer seine persönliche Glasnost im Umgang mit wenig bekleideten Menschen oder sexuellen Akten noch nicht vollzogen hat, der wird wahrscheinlich weniger Gefallen an den Taten des Larry finden, da dieser seine körperlichen Bedürfnisse innerhalb des Spiels nicht immer hinter zugezogenen Gardinen befriedigt. Damit kein falscher Verdacht (auch bei der Bundesprüfstelle für Jugendgefährdende Schriften) aufkommt: Bei dem dritten Teil von Leisure Suit Larry handelt es sich mitnichten um ein besonders sexualbezogenes oder gar pornografisches Werk, sondern um ein Stück erstklassige Unterhaltung für all diejenigen, die nicht so schnell mit Empörungskundgebungen bei der Hand sind. Man könnte sogar noch einen Schritt weitergehen und dem Spiel einen gewissen satirischen Touch nachweisen, denn: Nicht selten wird dem ach so modernen Menschen in diesem Programm der Spiegel der Selbstüberschätzung vorgehalten.

Um auf die Anfangsfrage zurückzukommen: Leisure Suit Larry III hat auch im dritten Teil nichts von seiner Anziehungskraft verloren. Die Qualität ist, im Vergleich zu seinen Vorgängern, sogar noch um ein paar Stufen gestiegen.

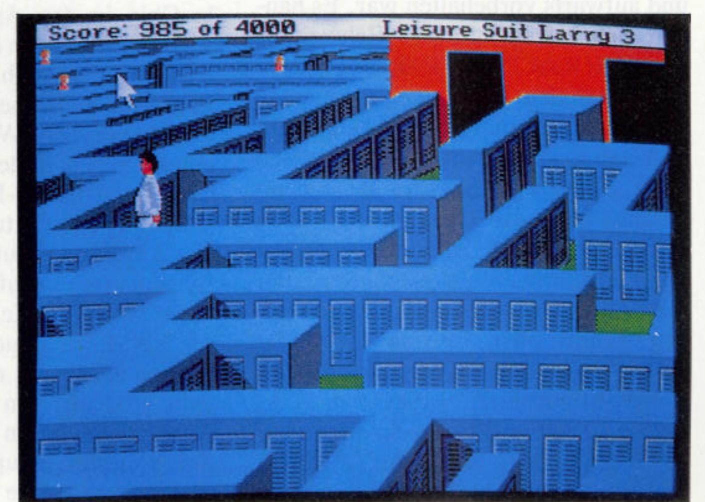
Hero's Quest dagegen weiß weniger Pluspunkte zu sammeln. Das hier begonnene neue Spielkonzept bedarf dringender Verbesserungen, bevor es eine ähnlich gute Note verdient.

Zwei Neue, einmal hui, einmal weniger hui, eine akzeptable Bilanz.

Hero's Quest – Leisure Suit Larry III
Preis: je 134,95 DM
Besonderheiten: nicht kopiergeschützt, unterstützen alle gängigen Sound- und Grafikkarten, wobei kein echter VGA-Modus existiert, Festplatte und schneller Prozessor werden dringend empfohlen

(jf)





Die Qualität der Larry-Serie steigt mit jeder neuen Folge
Screenshot: Leisure Suit Larry III



NVR durchleuchtet (1)

Zeit, RAuM und was sonst noch dazugehört

Die erste Begegnung mit dem kleinen Unterschied eines Amstrad-PCs zum Rest der normalen PC-Welt hat der Anwender gleich nach dem Aufstellen und Einschalten seines neuen Rechners. Nach der fröhlichen Meldung 'Bitte Warten' und dem Selbsttest des Rechners erscheinen bei einem neuen PC 1512/1640 noch folgende zwei Meldungen auf dem Bildschirm: Datum und Uhrzeit einstellen oder Standardeinstellungen ändern (wenn gewünscht). Der normale Anwender wird sich fragen: "Was hat das zu bedeuten?" und der Programmierwütige: "Was kann ich damit sonst noch anfangen?"

Auf beide Fragen wird hier nun versucht, eine befriedigende Antwort zu geben. Fangen wir mit dem einfachen an. Diese Meldungen kommen von einer Einrichtung, die früher (bis zu dem großen Markterfolg der Amstrad-Computer) Rechnern der AT-Klasse und aufwärts vorbehalten war. Es handelt sich hierbei um eine Echtzeituhr (englisch Real Time Clock / RTC), die die früher dem Anwender obliegende Aufgabe übernimmt, das Systemdatum und die Systemzeit auf den aktuellen Wert zu setzen (falls die RTC denn stimmt). Es gab zwar früher auch schon RTCs für PCs, doch existierten diese fast immer auf Slotkarten. Bei diesen handelt es sich aber meist um einen Chip der Firma National (MM58167), der im Vergleich zu den Chips (Motorola HD146818) in ATs, und den Amstrad-PCs, recht dumm ist. Er beherrscht keine Jahreswechsel, und Schaltjahre gibt es schon gar nicht. Andererseits hat er eine Zeitauflösung von einer tausendstel Sekunde und acht Interruptmöglichkeiten.

Das größte Problem ist aber meist das Fehlen jeglicher BIOS-Unterstützung der National Uhr und die Ignorierung der RTC durch das DOS. Der Motorola Chip in den Amstrads (und ATs) wird hingegen von deren BIOS voll unterstützt, und auch das DOS ignoriert ihn meist nicht mehr (siehe auch PCI 12/89 Seite 78). (Wenn von nun an von einer RTC die Rede ist, ist nur noch der Chip in Amstrad-Rechnern und ATs gemeint.) Ein Feature der RTC, das in den Amstrads benutzt wird, sind die 50 Bytes batteriegepufferten RAMs (englisch: Non Volatile Ram / NVR). Dieses RAM wird auch in ATs benutzt, dort natürlich mit einer anderen Belegung und anderen Aufgaben. In den Amstrad-Rechnern werden 26 Bytes des NVR für Setupzwecke des Rechners benutzt (siehe Abbildung 1). Diese Setup-Bytes lassen sich mit Hilfe des NVR-Programms auf der Systemdiskette ändern. Sie dienen zur Ablage allgemeiner Informationen, die das BIOS benötigt, um zum Beispiel die Sondertasten des Rechners richtig umzusetzen oder die serielle Schnittstelle zu initialisieren. 6 Bytes werden dazu verwendet, die Zeit der letzten Benutzung des Rechners abzuspeichern (die nette Meldung beim Kaltstart des Rechners). Über all diese Bytes (bis auf die, die etwas mit der Zeit zu tun haben) wird eine Checksumme berechnet. Falls beim Start des Rechners die Überprüfung des NVRs anhand der Checksumme fehlschlägt (Änderungen durch 'wildlaufende' Programme oder einfach nur leere Batterien), setzt das BIOS das NVR auf Standardwerte (Defaultspalte in der Abbildung 1) und bringt ebenfalls die am Anfang erwähnte Meldung.

Register und Speicher

Kommen wir allmählich zu den für Programmierer interessanten Einzelheiten der RTC. Die Abbildung 1 dient dazu, einen schnellen und möglichst kompletten Überblick über das NVR zu geben. Einige Bits und Bytes erfordern jedoch eine etwas genauere und damit längere Erklärung. Die Bytes 0 bis 9 sind eigentlich selbsterklärend. Es handelt sich abwechselnd um die aktuelle Zeit und eine mögliche Alarmzeit. Zu beachten ist lediglich, daß Byte 0 (Sekunden) sich nur lesen läßt und auch nur 7 Bits hat. Die darauf folgenden Bytes 10 bis 13 sind die Register A bis D der Uhr. Das Register A hat drei Aufgaben. Bit 7 (UIP) gibt an, wenn es gesetzt ist, daß der Chip gerade Zeit/Datum aktualisiert. Da dafür, durch etwaige Überläufe, etwas Zeit gebraucht wird, muß vor jedem Auslesen der Zeit anhand dieses Bits überprüft werden, ob die Werte gültig sind. Die Bits 6 bis 4 (DV2 - DV0) geben einen Teiler für eine der drei möglichen Betriebsfrequenzen an (Abbildung 2). Die Uhr wird normalerweise mit einer Frequenz von 32 KHz betrieben. Die beiden möglichen höheren Betriebsfrequenzen von 1 und 4 MHz haben zwar den Vorteil, daß die Uhr wesentlich genauer geht, aber auch wesentlich mehr Strom verbraucht. Von den möglichen acht Kombinationen dieser 3 Bits geben drei einen Teiler für die Betriebsfrequenz an, zwei lösen einen Reset der Teilerkette aus und halten dann die Uhr an, und die anderen drei dienen undo-

kumentierten Testzwecken von Motorola.

DV0	DV1	DV2	
0	0	0	4,194304 MHz
0	0	1	1,048576 MHz
0	1	0	32,768 KHz
1	1	0	Reset / Stop
1	1	1	“

Abbildung 2: Register A Bits 6 bis 4

Die Bits 3 bis 0 (RS3 bis RS0) stellen die Frequenz eines periodischen Ausgangssignals am Pin SQW des Chip und/oder eine periodische Interruptrate ein (Abbildung 3). Hierzu ist zu sagen, daß der Pin, ebenso wie CKOUT, SQW das periodische Ausgangssignal bei den Amstrad-PCs nicht mit der Hauptplatine verbunden ist (was einen erfahrenen Bastler jedoch nicht daran hindert ...).

Das Register B könnte man als 'Erlaube-' Register bezeichnen. Wenn das entsprechende Bit gesetzt ist, wird die zugeordnete Aktion erlaubt, ansonsten wird diese Funktion nicht ausgeführt. Bit 7 verhindert oder unterbricht, wenn gesetzt, ein Update der Zeit. Damit wird vermieden, daß die Uhr weiterläuft, wenn gerade eine neue Uhrzeit und/oder ein neues Datum eingestellt werden soll.

Die Möglichkeit, daß ein periodischer Interrupt ausgelöst wird, bleibt davon jedoch ausgenommen. Bit 6 (PIE) erlaubt einen periodischen Interrupt mit der von RS3 bis RS0 vorgegebenen Frequenz. Bit 5 (AIE) erlaubt dem Alarminterrupt aufzutreten, wenn die eingestellte Alarmzeit erreicht wurde. Bit 4 (UIE) ermöglicht einem sogenannten Update-Interrupt aufzutreten. Dieser Interrupt wird dann jedesmal ausgelöst, wenn die Uhr mit der Aktualisierung der Zeit fertig ist, das heißt also jede Sekunde. Bit 3 (SQWE) aktiviert den Ausgang des Rechteckgenerators (Pin SQW). Wie bereits weiter oben erwähnt, hängt dieser aber beim 1512/1640 'in der Luft'. Für Bastler sei noch anzumerken, daß damit auch der Stromverbrauch der Uhr drastisch ansteigt.

Bit 2 (DM) gibt an (= =0), ob die Zeitdaten als BCD-Ziffern (Binary Coded Decimal) – was Standard ist – oder (= =1) als binär kodierte Zahlen interpretiert werden. Bit 1 (24) entscheidet darüber, ob die Uhr im 24-Stunden-Betrieb (= =1) läuft oder die angelsächsische 12-Stunden-Methode (= =0) verwendet wird. Bei letzterer Methode entscheidet Bit 7 des Stundenbytes als am/pm Flag über Vor- oder Nachmittag. Bit 0 (DSE) schaltet die automatische Winter-/Sommerzeitanpassung ein. So praktisch das jedoch wäre, als Europäer kann man leider nichts damit anfangen, da die amerikanischen Termine von der Uhr zugrundegelegt werden.

Welcher Interrupt war's denn?

Über diese Read-Only-Register der Uhr läßt sich feststellen, welche Art von Interrupt durch die Uhr ausgelöst wurde. Durch 'geschickte' Programmierung ist es ohne weiteres möglich, daß alle drei Interrupts der Uhr zugleich auftreten. Ein Problem ist, daß jegliches Lesen dieses Registers alle Interruptanforderungen löscht. Darauf muß man in seiner Behandlungsroutine für Uhrinterrupts achten, wenn man kein mittleres Chaos auslösen will. Bit 7 (IRQF) ist gesetzt, wenn einer der drei möglichen Interrupts aufgetreten ist. Bit 6 (PF) ist gesetzt, wenn ein periodischer Interrupt, Bit 5 (AF), wenn ein Alarm-Interrupt und Bit 4 (UF), wenn ein Update-Interrupt aufgetreten ist. Der Rest der Bits des Registers bleibt immer auf 0 gesetzt. Beim letzten Register, D (byte 13), ist nur ein einziges Bit benutzt. Dieses Bit, es handelt sich um das siebte, ist gesetzt, wenn die Spannung der Uhr in Ordnung ist. Daher weiß das BIOS auch, wann es die Meldung 'Bitte neue Batterien einsetzen' bringen muß. Der Rest dieses Read-Only-Registers ist schlicht 0. Nach einem Lesevorgang wird das Bit automatisch auf 1 gesetzt.

Der Rest

In den Bytes 14 bis 19 merkt sich der Rechner, wann er zuletzt benutzt wurde. Das darauf folgende Byte dient als Ausgleich für die Checksumme des NVRs. Diese wird jedesmal bei einem Systemstart berechnet. Tritt ein Fehler auf, wird das NVR auf Defaultwerte gesetzt, und die schon erwähnte Meldung wird ausgegeben. Die Berechnung dieser Checksumme geht folgendermaßen vor sich: Alle Bytes zwischen 20 und 63 werden zusammengezählt. Das Low-Byte dieses Ergebnisses muß 0AAH sein, ansonsten wird die Checksumme als fehlerhaft angese-

RS3	RS2	RS1	RS0	Frequenz in Hz
0	0	1	1	8192
0	1	0	0	4096
0	1	0	1	2048
0	1	1	0	1024
0	1	1	1	512
1	0	0	0	256
1	0	0	1	128
1	0	1	0	64
1	0	1	1	32
1	1	0	0	16
1	1	0	1	8
1	1	1	0	4
1	1	1	1	2

Abbildung 3: Periodische Interrupts

Byte	Verwendung	Default	Anmerkung
0	Sekunde / Uhrzeit	0	Nur 7 Bit (r/o)
1	" / Alarm	0	
2	Minute / Uhrzeit	0	
3	" / Alarm	0	
4	Stunde / Uhrzeit	0	
5	Stunde / Alarm	0	
6	Tag der Woche	3	Sonntag == 1, ...
7	Tag des Monats	1	1 bis 31
8	Monat	1	Januar == 1, ...
9	Jahr	80	Als Jahrhundert wird 19 oder 20 angenommen
10	Register A (r/w)	70H	BIT Verwendung 7 UIP Update in Arbeit (wenn gesetzt) 6 DV2 Teiler für Betriebsfrequenz 5 DV1 " 4 DV0 " 3 RS3 Teiler für Interrupten 2 RS2 " 1 RS1 " 0 RS0 "
11	Register B (r/w)	02H	BIT Verwendung (wenn gesetzt) 7 SET Update nicht erlaubt 6 PIE Periodische Interrupts erlaubt 5 AIE Alarm Interrupt erlaubt 4 UIE Update Interrupt erlaubt 3 SQWE Rechteckausgang aktiviert 2 DM Binäre Daten statt BCD 1 24 24 Stunden statt 12 (am/pm) 0 DSE Sommer/Winterzeit (leider Eng.)
12	Register C (r/o)		BIT Verwendung (gesetzt wenn) 7 IRQF Interrupt ist aufgetretn 6 PF Periodischer Interrupt 5 AF Alarm Interrupt 4 UF Update Interrupt 3-0 Immer 0
13	Register D (r/o)		BIT Verwendung (gesetzt wenn) 7 VRT Spannung ok 6-0 Immer 0
14-19	Zeit und Datum		Zeit und Datum der letzten Benutzung
20	Checksumme		Checksumme des NVR_Rams (s. Text)
21-22	Enter Taste	1C0DH	Übersetzung für Entertaste (NumPad)
23-24	Del-> Taste	2207H	Übersetzung für Del-> Taste (Crtl G)
25-26	Joystick Taste 1	FFFFH	Übersetzung für Joystickfeuer 1 *
27-28	Joystick Taste 2	FFFFH	" 2 *
29-30	Linke Maustaste	FFFFH	Übersetzung der Linken Maustaste *
31-32	Rechte Maustaste	FFFFH	" *
33	Maus X Faktor	0AH	Bewegungsmaßstab in X-Richtung
34	Maus Y Faktor	0AH	Bewegungsmaßstab in Y-Richtung
35	Einschaltmodus/ Laufwerke	20H	Art des Textmodus beim Einschalten und Anzahl der Diskettenlaufwerke BIT Verwendung 7 wenn gesetzt I/O Control Option ein 6 1 wenn zwei Laufwerke sonst 0 5 Videomode 4 " mögliche Kombinationen 01 Farbe, Text 40*25 10 Farbe, Text 80*25 11 Externer Monochromcontroller 3-0 0
36	Einschaltattribut	07H	Textattr., default ist weiß auf schwarz
37	Größe der Ramdisk	00H	Größe der Ramdisk in 2 kByteschritten. Wird nur vom mitgelieferten MS-DOS 3.2 RAMDISK.SYS und DOSPLUS verwendet.
38	Standard RS-232 (intern)	E3H	Standardinitialisationswert für die interne serielle Schnittstelle BIT Verwendung 7-5 Baudrate mögliche Kombinationen 000 110 Baud 001 150 Baud 010 300 Baud 011 600 Baud 100 1200 Baud 101 2400 Baud 110 4800 Baud 111 9600 Baud 4 gesetzt even Parity, sonst odd Parity 3 wenn gesetzt Parity möglich 2 gesetzt 2 Stop Bits sonst 1 Stop Bit 1 immer gesetzt 0 gesetzt 8 Data Bits sonst 7 Data Bits Der Default Wert ergibt 9600 Baud, 8 DATA Bits, 1 Stop Bit, keine Parität
39	Standard RS-232	E3H	Siehe Standard RS-232
40-63	Nicht benutzt		Diese Bytes werden vom Bios nicht benutzt Sie stehen dem Benutzer zur Verfügung.

Abb.1: Speicherstellen im NVR

* Der Wert FFFFH hat eine spezielle Bedeutung. Bei diesem Wert wird die Taste vom Treiber einfach ignoriert.
Alle Bytes und Register, bei denen in der Spalte Default kein Wert angegeben wird, werden vom Bios bei Fehler im NVR nicht angetastet.

hen. Um auf diesen Wert zu kommen, wird ein Ausgleichswert im Byte 20 verwendet. Nach diesem Checksumbyte folgen 12 Bytes, die jeweils in Zweier-Gruppen den Scancode einer der Spezialtasten (wie Joystick-, Maus- oder Enter-und Del-Taste) des Rechners enthalten. Wird eine dieser Tasten betätigt, nimmt der dafür zuständige Treiber anhand dieser Werte eine Umkodierung der Taste vor. Zu beachten ist hierbei, daß ein Wert von -1 (hexadezimal 0FFFFH) dazu führt, daß die Taste vom Treiber einfach ignoriert wird. Die beiden nächsten Bytes (33, 34) geben einen Umrechnungsfaktor für die Bewegung der Maus an. Wie der Benutzer eines Amstrad-PCs weiß, wird die Bewegung der Maus, falls das gerade laufende Programm keine Mausunterstützung erlaubt, automatisch in Cursorbewegungen umgesetzt. Diese beiden Werte geben jeweils an, wie viele interne Bewegungseinheiten der Maus in eine Cursorbewegung, getrennt in horizontale und vertikale Bewegung, umgesetzt werden. Bytes 35 und 36 dienen dazu, den Standardmodus und die Standardfarben des Bildschirms beim Einschalten des Rechners festzulegen. Da bei Byte 35 (für den Einschaltmodus zuständig) noch einige Bits überbleiben, wird Bit 7 dafür verwendet, bei der seriellen Schnittstelle den Handshake bei der Kopplung von zwei Rechnern an- oder abzustellen. Bit 6 gibt an (wenn gesetzt) ob zwei Laufwerke im System sind. Die Bits 5 und 4 geben, wie gesagt, den Bildschirmmodus an. Die Bits 3 bis 0 werden nicht verwendet. Das Byte 37 gibt die Größe der RAM-Disk in 2-kByte-Schritten an. Dieser Wert wird jedoch nur von DOS-Plus und von dem RAM-Disk-Programm (hier auch nur, wenn der Parameter NVR benutzt wird) des von Amstrad mitgelieferten MS-DOS 3.20 benutzt. Alle anderen RAM-Disk-Programme können mit diesem Wert nichts anfangen. Wer also eine andere als die originale DOS-Version benutzt, kann dieses Byte als ziemlich unnütz betrachten. Die letzten beiden vom BIOS belegten Bytes (38 & 39) dienen dazu, die Initialisierungswerte für die interne (Byte 38) und eine externe (Byte 39) serielle Schnittstelle aufzunehmen (Einzelheiten in Abbildung 1). Für alle Programmierer nun die gute Nachricht. Die Bytes 40 bis 63 sind frei und stehen dem Benutzer zur absolut freien Verfügung. Wie man nun auf all diese Feature der Uhr, respektive des NVRs, zugreift, das alles wird in der nächsten Folge genauer erklärt.

(Robert Haas/jf)

MS-DOS, die ersten Berührungen (2.Teil)

Die AUTOEXEC.BAT-Datei und andere Stapel



Daß Computer stark vereinfachte Sprachen zur Befehlseingabe verlangen, liegt in der Sache begründet. Für den Programmierer ist es wesentlich einfacher, wenn dieser den Befehlssatz erheblich einschränkt, da er die Eingabe dann entsprechend strukturiert behandeln kann.
Das Zaubermotto heißt daher "Kurz gefaßt und abgekürzt". Gerade in der simplen Namensgebung von Dateien und Programmen findet sich dieses Motto in besonderem Maße wieder, da die Länge der Programmnamen auf elf Buchstaben beschränkt sind.

Die Amerikaner spielen in den Namen besonders gerne mit ihren Zahlen "2" und "4" herum. Nutzt man diese in Dateinamen, so lassen sich durch die sprachliche Doppeldeutigkeit der Ziffern (2 = t(w)o - hin, zu / 4 = fo(u)r - für, zu diesem Zweck) auf den ersten Blick unsinnige, beim zweiten Hinhören dann aber interessante Namensbildungen schaffen:
4BILL.DAT - Daten für Bill
EXE2BIN.COM - EXE-Datei wird zur BIN-Datei
Die meisten Dateien bestehen dann aber doch aus wesentlich einfallsloseren Abkürzungen, die Personen ohne große Englischkenntnisse aber schon für sehr obskure Gebilde ansehen müssen.
Die AUTOEXEC.BAT-Datei etwa scheint auf den ersten Blick den Verdacht zu erwecken, ein Scherzbold sondergleichen habe hier eine Idee verwirklicht, die den Computernutzern außer viel Schreibarbeit wenig glückliche Momente bringt. Schaut man aber genauer hin, entpuppt sich das Buchstabengewirr als eine Anhäufung von Informationen. Der Begriff "autoexec" ist dabei ein Abkürzung von "auto execute", was übersetzt soviel heißt wie "selbständig ausführend". Die Dateikennung "bat" steht für Batch-Datei (Stapel-Datei), eine Datei, die eine Ansammlung von Befehlen enthält und, sofern aufgerufen, diese der Reihe nach ausführt.

Erst mal schaun, was AUTOEXEC hat...

Diese Stapeldatei trägt den Namen "selbständig ausführend" nicht umsonst. Unser Betriebsystem ruft diese Datei dann auch nach dem "Booten" eigenhändig auf, sofern sie vorhanden ist (das schrittweise Vorgehen des Computers beim Start sehen Sie in Abbildung 1), und veranlaßt die Abarbeitung aller darin enthaltenen Befehle. Der Anwender kann dies zu seinem Vorteil nutzen, indem er diese Datei so einrichtet, daß sie bestimmte Arbeiten verrichtet. Dazu steht ihm ein Befehlssatz zur Verfügung, der wie eine eingeschränkte Programmiersprache behandelt werden kann (Abbildung 2). So klein dieser Befehlssatz auch wirken mag, so fantastische Ergebnisse können bei geschickter Anwendung damit erreicht werden.

Zu diesen speziellen Stapelverarbeitungsbefehlen können weiterhin alle denkbaren MS-DOS-Befehle gebraucht werden. Dabei ist es egal, ob es sich um ein residentes oder um ein transientes Kommando handelt. Ein residenter Befehl steht dem Kommandointerpreter stets zur Verfügung (DIR, TYPE...), wogegen ein transienter Befehl ein auf Datenträger gelagertes Programm ist, was vor der Abarbeitung erst einmal geladen werden muß (XCOPY, COMP...).

Schließlich ist es denkbar, alle startbaren Programme aus einer Batch-Datei aufzurufen.

Außer den Aufrufmöglichkeiten der Batch-Dateien, können diesen auch bestimmte Parameter mitgegeben werden, die dann innerhalb der Batchdatei für variables Arbeiten sorgt.
Was aber ist nun das eigentlich besondere an dieser Datei, außer daß MS-DOS sie automatisch nach dem Systemstart abarbeitet?

Nicht nur für Einsteiger...

Der erste große Pluspunkt ist natürlich, daß aus einer solchen Datei eine beliebige Anwendung mit den dazu eventuell sinnvollen speicherresidenten Hilfsprogrammen ohne jegliches Dazutun des Anwenders geladen werden kann. Für Einsteiger oder Personen, die nichts anderes mit ihrem Computer zu tun gedenken, als bestimmte Anwendungen zu fahren, kann eine solche Datei eine Menge Arbeit und Einarbeit sparen. Er braucht keinerlei Kenntnisse von Betriebssystemen, Programmnamen oder ausführbaren Dateien zu haben, die AUTOEXEC.BAT-Datei erledigt alles für ihn.

1. MS-DOS wird gebootet
2. Existiert die CONFIG.SYS-Datei, so werden die darin enthaltenden Konfigurationen eingestellt. Existiert keine, so greift das System auf die Standardeinstellung zurück.
3. COMMAND.COM wird geladen
4. Existiert die AUTOEXEC.BAT-Datei, so wird sie ausgeführt, andernfalls werden Datum und Uhrzeit vom System abgefragt.
5. Freigabe für Eingaben des Benutzers

Abb. 1: Startmanöver des Computers

Für die Personengruppen, die sich ihren Computer selbst einrichten wollen, hat die AUTOEXEC.BAT-Datei einen ähnlichen Effekt. Jegliche Voreinstellungen vom Prompt bis zum Grafikmodus können gleich beim Start erledigt werden, ohne daß der Benutzer sich in zeitraubende Tipparbeit zu stürzen hat. (Sie erinnern sich an unsere erste Folge, wo wir die tippintensive Promptdefinition in einer Batch-Datei untergebracht hatten.)

Batch-Dateien sind also die perfekte Unterbringung von häufig wiederkehrenden Befehlsfolgen oder Anwendungsinstallationen. Außerdem eignen sie sich sehr gut dazu, kleinere Aufgaben zu erledigen. (Näheres dazu im kleinen Programmbeispiel am Ende dieser Folge.)

Probleme mit Stapeldateien

Bei all dieser Flexibilität sind natürlich bestimmte Vereinbarungen einzuhalten. Bei der Wahl des Namens kann es so durchaus vorkommen, daß der Anwender eine Batch-Datei schreibt, die trotz korrekten Aufrufes nie abgearbeitet wird. Dazu muß man wissen, daß das Betriebssystem MS-DOS bestimmte Prioritäten bei dem Programmaufruf besitzt. Es richtet sich dabei nach den Dateikennungen, die wir bereits in der ersten Folge durchleuchtet haben. Wir erinnern uns, daß im Normalfall nur Dateien mit der Kennung .COM, .EXE oder .BAT ausgeführt werden. Besitzen nun Dateien die gleichen Namen, aber jeweils eine andere der oben genannten Kennungen, so muß sich das Betriebssystem entscheiden, welcher Datei der Vorrang gegeben wird. Die vereinbarte Reihenfolge ist dabei:

1. COM-Datei (command)
2. EXE-Datei (execute)
3. BAT-Datei (batch)

Sollten Sie also eine Batch-Datei mit Namen Such.bat erstellt haben und sich zufälligerweise ein Programm namens Such.exe auf dem Datenträger befinden, so wird nach dem Start (such <ENTER>) stets die .EXE-Datei aufgerufen, niemals aber die Batch-Datei. Neuere DOS-Versionen erlauben mittlerweile zwar den Aufruf >Such.bat<, der sicherstellt, daß tatsächlich auch die Batch-Datei behandelt wird, doch sollte man sich selbst aus solchen zwielichtigen Situationen fernhalten, indem man gleiche Namensgebungen umgeht.

Call
Der Befehl CALL dient zum Aufruf einer Batch-Datei aus einer solchen heraus. Diese Aufrufe können beliebig verschachtelt werden, der Speicherplatz dient dabei als natürliche Grenze.
Aufruf:
Call <Pfad \ Stapeldateiname \ Parameter>

ECHO
Zwei Funktionen werden diesem Befehl zugeschrieben. Die eine ist eine Schalterfunktion und ermöglicht es dem Benutzer, das Schreiben des momentanen Befehls auf den Bildschirm zu unterbinden.
Aufruf:
ECHO ON/OFF
Mit der zweiten Funktion können beliebige Kommentare oder Aufforderungen auf den Bildschirm geschrieben werden, wobei ein einfacher Punkt (.) das Schreiben einer Leerzeile verursacht.
Aufruf:
ECHO <Text>

FOR - IN - DO
Diese Schleife sorgt für das mehrmalige Abarbeiten bestimmter Befehle mit wechselnden Bedingungen, wobei eine Verschachtelung der Schleifen nicht erlaubt ist. Bei dieser Art von Schleife nimmt eine Variable nacheinander jeden Wert einer Menge an. Die Menge kann dabei aus beliebigen Elementen (auch Dateinamen mit Wildcards) bestehen.
Aufruf:
FOR %%<Variable> IN <Menge> DO <Befehl>

GOTO
Der Befehl GOTO veranlaßt einen Sprung zu einem mit :<Name> gekennzeichneten Label. Durch diesen Befehl ist es möglich, die sonst nur geradlinige Abarbeitung von Stapeldateien zu unterbrechen und beispielsweise auch Rücksprünge oder schleifenähnliche Konstrukte zu ermöglichen.
Aufruf:
GOTO <Name>

IF
Mittels IF-Kommando lasen sich drei unterschiedliche Bedingungen erfragen:
1. exist <Dateiname>
2. <Wort1> == <Wort2>
3. errorlevel <Fehlerwert>
Exist prüft den angegebenen Datenträger auf Vorhandensein der angegebenen Datei
Der zweite Fall ist ein Stringvergleich, der es Ihnen ermöglicht, Worte auf Gleichheit zu überprüfen.
Die Errorlevel-Abfrage kontrolliert bei auftretenden Fehlern die von einer Anwendung zurückgegebenen Fehlerwerte.
Aufruf:
IF exist <Dateiname> <Befehl>
IF <Wort1> == <Wort2> <Befehl>
IF errorlevel <Fehlerwert> <Befehl>

PAUSE
Pause sorgt für eine Unterbrechung der Stapelabarbeitung. Der Befehl kann mit einem Text versehen werden. Wird dieses unterlassen, so erscheint eine Standardmeldung auf dem Bildschirm, die durch eine Umleitung ins "Nirwana" vermieden werden kann (Abbildung 3).
Aufruf:
PAUSE <text>
PAUSE >NUL

REM
Markiert in einer Batchdatei eine Kommentarzeile, die nicht bearbeitet wird.
Aufruf:
REM <Text>

SHIFT
Durch den Befehl Shift wird die Liste der an die Batch-Datei übergebenen Parameter um eine Stelle verschoben. %0 wird %1, %1 wird %2, %2 wird %3 usw. Durch diese Aktion wird der höchste Parameter %9 freigesetzt und kann neu benutzt werden.
Aufruf:
SHIFT

Abb. 2: Batch-Befehle

Ein weiteres Problem, das man schnell mit Batch-Dateien bekommen kann, ist der Datenträgerwechsel. Dazu muß die Vorgehensweise einer solchen Stapelverarbeitung betrachtet werden.

Der Kommandointerpreter, also der Teil des Betriebssystems, der auf Ihre Eingaben von der Tastatur wartet und diese zu entsprechender Ausführung bringt, ändert seine Arbeitsweise prompt, sobald eine Batchdatei gestartet wird. Dann weiß er, daß er ab jetzt nicht auf die Eingabe eines Kommandos per Tastatur zu warten hat, sondern daß er Schritt für Schritt die Befehle vom Befehlsstapel der Batchdatei zu nehmen hat. Das tut er im übrigen solange, bis er von der Stapeldatei durch Übergabe von CTRL-Z gesagt bekommt, daß die Stapelverarbeitung nun beendet ist und daß der Kommandointerpreter in den normalen Betriebszustand überwechseln kann. Da der Kommandointerpreter nun aber nicht die gesamte Stapeldatei in den Arbeitsspeicher lädt, sondern Zeile für Zeile anfordert und abarbeitet, kommt ein Problem auf:

Die Batch-Datei will ein Programm laden, das auf einem anderen, als dem momentanen Datenträger vorhanden ist.

Da das aufgerufenen Programm nach Ende wieder an den Kommandointerpreter zurückgibt, dieser aber noch in der Abarbeitung der aktuellen Batch-Datei ist, versucht er selbstverständlich, die nächste Befehlszeile aus der Batch-Datei anzufordern. Sollte sich nun nicht die richtige Diskette im Laufwerk befinden, so kommt es zu einer Fehlermeldung. Um diese zu umgehen, kann man mit dem PAUSE-Kommando an geeigneten Stellen die Abarbeitung der Stapeldatei unterbrechen (beispielsweise für den Diskettenwechsel).

Will man generell eine Batch-Datei unterbrechen, so kann man das durch gleichzeitiges Drücken der Tasten "CTRL" und "C" erreichen.

Das Errichten einer Stapel-Datei

Wir kennen jetzt die Befehle und Möglichkeiten der Batch-Dateien, einige Probleme mit diesen und wie man mit ihnen umgeht. Was fehlt, ist das Errichten einer solchen Datei. Wie immer sind wir auch hier nicht auf eine bestimmte Vorgehensweise festgelegt. Längere Batchdateien werden zweckmäßigerweise mit einem beliebigen Texteditor geschrieben und als ASCII-File auf Datenträger gesichert (denken Sie an die Kennung .BAT).

<Dateiname>	– Umleitung in eine Datei mit angegebenen Namen.
NUL	– Umleitung ins "Nirwana". Diese Daten oder Meldungen gelangen in den großen Mülleimer des MS-DOS und werden artgerecht entsorgt.
PRN	– Direktumleitung auf den Drucker
LPT	– Senden der Daten zur parallelen Schnittstelle
COM	– Senden der Daten zur seriellen Schnittstelle

Abb. 3: Dateiumleitungen

Bei kürzeren Stapeldateien bietet MS-DOS mit dem COPY CON-Befehl eine schnelle Möglichkeit, kleine Dateien direkt von der Tastatur auf Diskette oder Festplatte zu schreiben. Wir haben eben diesen in der ersten Folge bereits dazu verwendet, die Promptdefinition zu sichern. Durch

COPY CON <Dateiname.BAT>

sind wir in der Lage, Zeile für Zeile eine Stapeldatei zu füllen. Zeile für Zeile deshalb, weil diese Art der Eingabe der eines Zeileneditors ähnlich ist. Das heißt, eine einmal durch ENTER bestätigte Zeile kann nicht ohne weiteres verbessert oder verändert werden. Bei dieser Eingabeform ist also auf korrekte Schreibweisen der einzelnen Befehle zu achten. Wenn wir alle unsere Befehle eingegeben haben, vermitteln wir dem Kommandointerpreter durch Eingabe von CTRL-Z (die Taste F6 enthält diese Zeichenfolge), daß wir fertig sind. CTRL-Z gibt dem Kommandointerpreter also nicht nur bei der Abarbeitung an, daß eine Stapeldatei beendet ist, sondern auch bei der Eingabe.

Ein Beispiel

Als kleines Beispiel für eine sinnvolle Aufgabe einer Batch-Datei soll folgendes kleines Programm dienen. Es handelt sich dabei um eine Batch-Datei, die alle Dateien eines Datenträgers, auch die in dessen Unterverzeichnissen, nach einem vom Benutzer eingegebenen Kennwort durchsucht und alle Dateien, die dieses Kennwort im Namen besitzen, in eine Datei namens FUND.NEU schreibt und danach auf dem Bildschirm ausgibt. Wenn sich bereits eine Datei namens FUND.NEU auf dem Datenträger befindet, so wird sie in FUND.ALT umbenannt und eine eventuelle Datei namens FUND.ALT gelöscht.

(Diese kleine Batch-Datei ist auch auf der DATABOX enthalten.)

```
echo off
cls
echo.
echo.
echo ***************************
echo *  Bitte Taste druecken, um die  *
echo *          Suche zu starten         *
echo *                                   *
echo *       Abbruch mit CTRL-C          *
echo ***************************
echo.
pause > nul
if exist fund.alt del fund.alt
if exist fund.neu ren fund.neu fund.alt
chkdsk /v |find "%1" > fund.neu
cls
type fund.neu
```

Wie Sie sehen, nutzt diese Batch-Datei einige Besonderheiten des Befehlssatzes der Stapelverarbeitung aus.

Das Größer-Zeichen (>) zum Beispiel dient dabei zur Umleitung bestimmter Meldungen oder Daten (Abbildung 3).

Um diese Batchdatei zu nutzen, muß sie in richtiger Weise aufgerufen werden. Wie Sie sehen, benutzt die Datei eine Variable (%1), deren Inhalt wir der Batch-Datei beim Start als Parameter übergeben müssen. Um also beispielsweise alle Dateien suchen zu lassen, die im Namen die Buchstabenfolge BAT besitzen, geben wir ein:

such BAT

Die Großschreibung ist nötig, weil die Buchstaben der Dateinamen groß geschrieben sind.

Ausblick

Nachdem wir nun einen Überblick über die Stapelverarbeitung und deren wichtigsten Vertreter, die AUTOEXEC.BAT-Datei, bekommen haben, wollen wir uns in der nächsten Folge mit einem weiteren wichtigen Datei-Vertreter beschäftigen, jener Datenanhäufung, die die Voreinstellung der Systemkonfiguration vornimmt: CONFIG.SYS.

(jf)



JR in die Karten geschaut

Das Strategiespiel For President

Die ersten Ergebnisse unserer Leserumfrage haben es an den Tag gebracht: Die Mehrzahl unserer Leser würde es begrüßen, wenn wir besonders lange Listings im Heft ausführlich vorstellten, das eigentliche Listing dann aber nicht abdruckten, sondern auf der DATABOX herausbrächten. Das hat natürlich den Vorteil, daß wir kostbaren Heftplatz für weitere Artikel nutzen können.
Wir wollen in dieser Ausgabe zum ersten Mal ein Programm auf diese Weise behandeln und wären natürlich froh darüber, wenn Sie uns dazu Ihre Meinung schreiben würden.

Amerika in den 60er Jahren, die Zeit des wirtschaftlichen Aufschwungs. Vor allem wegen des reichen Ölvorkommens werden viele Menschen wohlhabend. Es ist aber auch der Aufbruch ins Zeitalter der Raumfahrt. Viele Ergebnisse werden erzielt, die sich wirklich sehen lassen können.
Die 60er Jahre brachten jedoch leider nicht nur Wohlstand, sondern auch Armut, Rassenkrawalle und starke politische Auseinandersetzungen mit sich.

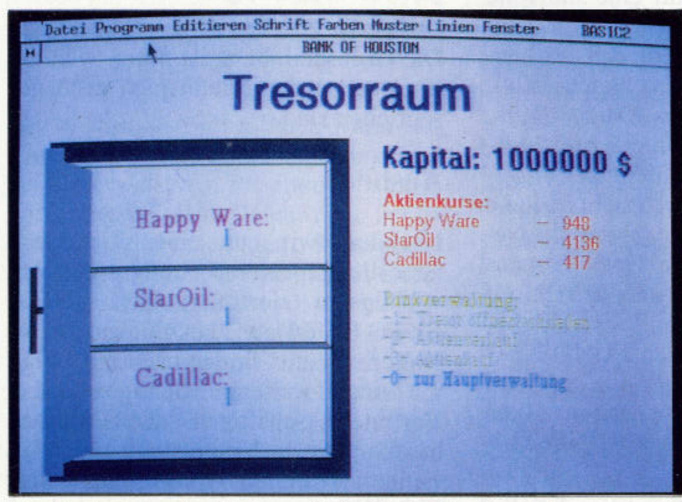
Es ist nicht immer leicht, Präsident zu sein. Martin Luther Kings Erfolg war leider nicht von Dauer, und selbst US-Präsident John F. Kennedy konnte nicht das erreichen, was er wollte.
Nun sind Sie am Zug. Sie wollen nur eins: Präsident der Vereinigten Staaten werden. Um dies zu erreichen, muß man natürlich starken politischen Einfluß besitzen und auch die Herzen der Bürger für sich gewinnen. Darum müssen Sie auf sich aufmerksam machen und in Fernsehen und Rundfunk für sich werben.
Die Politik ist ebenfalls mit enormen Kosten verbunden. Dieses Geld können Sie aufbringen, indem Sie mit Aktien handeln, nach Öl bohren und Plantagen bepflanzen. Jedoch sind Sie nicht der einzige Präsidentschaftskandidat, es gibt noch viele andere neidische Konkurrenten, die Ihnen das Leben schwer machen können. Aus diesem Grund ist es empfehlenswert, Ihre Ölfelder zu schützen und Agenten für diesen Zweck einzustellen. Doch ist nichts vorhersehbar, und das Leben spielt immer wieder den einen oder anderen Streich…

Das Spiel

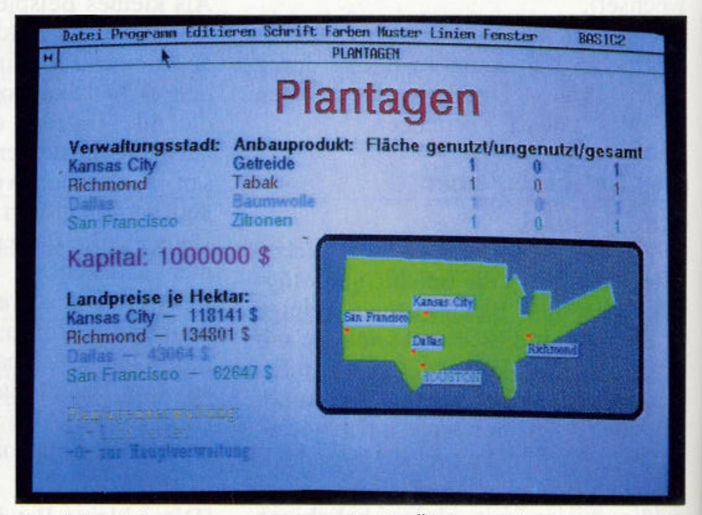
Die Steuerung des Spiels erfolgt über Menüs, um den Ablauf zu erleichtern. Nach den 'Vorbereitungen' gelangt man ins Hauptmenü, das sich in folgende Unterpunkte aufteilt:
Aus der Hauptverwaltung in Houston sind folgende Aktionen durchzuführen:

1 – HANDEL EINKAUFS/VERKAUFS-VERWALTUNG – Markt

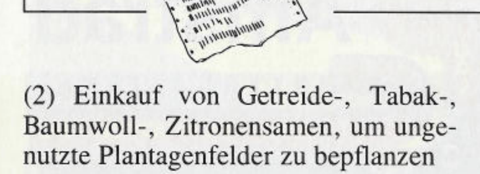
(1) Verkauf von Öl, Getreide, Tabak, Baumwolle und Zitronen



Die Manipulation der Börse geschieht aus dem Bankmenü heraus



Land, das nicht genutzt wird, ist gut für die Ökologie, aber schlecht für die Geldbörse; hier zählen aber ausschließlich die Finanzen



(2) Einkauf von Getreide-, Tabak-, Baumwoll-, Zitronensamen, um ungenutzte Plantagenfelder zu bepflanzen

(0) zur Hauptverwaltung

2 – BANK
BANKVERWALTUNG – Tresorraum

Sie sehen die Aktienkurse, und wenn Sie den Tresor öffnen auch die Anzahl Ihrer Aktien. Sie können von jeder Aktiengesellschaft höchstens 5000 Aktien besitzen.

(1) Tresor öffnen/schließen
(2) Aktienverkauf
(3) Aktienkauf
(4) zur Hauptverwaltung

3 – ÖL
ÖLVERWALTUNG – Ölbohrland in Dallas

Es werden die Anzahl der Öltürme angezeigt (bis zu 10), die Bohrtiefe je Bohranlage und das Volumen des Tanks. Sie sehen auch durch eine zeichnerische Darstellung, ob eine Pipeline bzw. Reinigungsanlage vorhanden ist. Ist beispielsweise keine Raffinerie auf dem Bild, müssen Sie eine neue kaufen, damit der 'Öllauf' weitergehen kann. Sie bekommen pro Monat höchstens soviel Öl, wie Ihr Tank aufnehmen kann.

(1) Gestängekauf

Hier können Sie Gestänge zu je 10-Meter-Stücken kaufen. Die maximale Bohrtiefe beträgt 2000 Meter (200×10m-Stücke).

(2) Tankerweiterung

Das Volumen des Tanks kann erhöht werden (max. 20 Millionen Liter)

(3) Kauf

Kauf einer Bohranlage (max. 10), Pipeline (max. 1), Reinigungsanlage

(4) Anlagenverkauf

Verkauf einer Bohranlage zum halben Einkaufspreis

(0) zur Hauptverwaltung

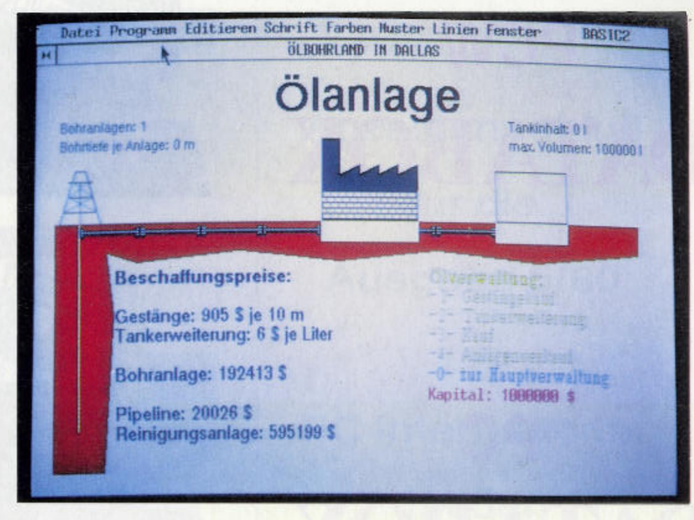
4 – PLANTAGEN
PLANTAGENVERWALTUNG – Plantagen

Hier können Sie die Größe der Plantagenfelder, Ihre Verwaltungsstadt, die genutzte und die ungenutzte Fläche sehen. Die ungenutzten Felder können durch den Kauf von Samen bepflanzt werden. (z. B.: 4 ha ungenutzte Felder in Richmond; Einkauf von 4 cwt Tabaksamen = 0 ungen. Felder).

(1) Land kaufen

Kauf von Plantagenfeldern in Kansas City, Richmond, Dallas, San Francisco (je Verwaltungsstadt höchstens bis zu 100 Hektar)

(0) zur Hauptverwaltung



Wer tief bohrt, der wird meist fündig. Daß diese Weisheit auch im Ölgeschäft einen wahren Kern hat, werden Sie selbst schnell merken

5 – POLITIK / SABOTAGE
ÜBERSICHT

Durch Drücken der Taste <0> können Sie jeweils aus den Untermenüs zurück zum nächsthöheren Menü.

(1) Politik, Werbung – Politische Verwaltung

a) Politisches Engagement

I. Innenpolitische Ausgaben
II. Außenpolitische Ausgaben

Folgende Mindestausgaben sind vorgeschrieben:
Innenpolit.: 1 Mill. Dollar
Außenpolit.: 1.5 Mill. Dollar

b) Werben von Bürgern mit Hilfe von: TV, Radio, Plakate

Diese Werbemaßnahmen nehmen Einfluß auf Ihre Beliebtheit.

(2) Arbeiter und Angestellte – Personalverwaltung

a) Einstellen von Arbeitskräften

Sie können maximal 800 Arbeiter einstellen, wobei für jeden Kosten in Höhe von $ 900 aufzubringen sind. Die maximale Anzahl von 50 Aufpassern verlangt pro Kopf $ 1500. Agenten sind maximal 10 anzuheuern, wobei jeder die stolze Summe von $ 12000 verlangt.

b) Entlassen von Arbeitskräften

Mindestens ein Plantagenarbeiter muß immer unter Vertrag stehen.

(3) Sabotageakt

Hier können Sie, sofern mehr als ein Kandidat am Spiel beteiligt ist, nach Wahl des Gegners und des ausführenden Saboteurs einen Sabotageakt begehen lassen. Wenn Sie Glück haben, bekommen Sie eine Beute, und der Gegner verliert seine Pipeline oder Raffinerie oder beides. Falls Sie jedoch Pech haben sollten, kann Ihr Saboteur noch entkommen, und es passiert gar nichts.

Wenn Sie erwischt werden, fällt Ihr Ansehen in der Öffentlichkeit (=Beliebtheit), und außerdem zahlen Sie Ihrem Gegner ein Bußgeld. Zum Schutz gegen Sabotageakte stellt man Aufpasser und Agenten ein.

0 – WEITER

Es erscheint ein Zufallsereignis, am Monatsende ein historisches Ereignis und eventuell ein Hilfeereignis. Nach Tastendruck ist der nächste Spieler an der Reihe.

Sie können den Spielstand mit <s> abspeichern, wenn die entsprechende Meldung erscheint: '(s) Save -(Taste) nächster Kandidat'.

Die Startvorgaben:
- $ 1.000.000 Kapital
- je eine Aktie jeder Gesellschaft eine Bohranlage (incl. Pipeline und Raffinerie)
- ein Tankvolumen von 100.000 l
- 1 ha beflanztes Land
- 1 Arbeitskraft

Allgemeine Hinweise

Das Programm läuft unter GEM Basic2 auf einem EGA-Monitor und entsprechender Karte.

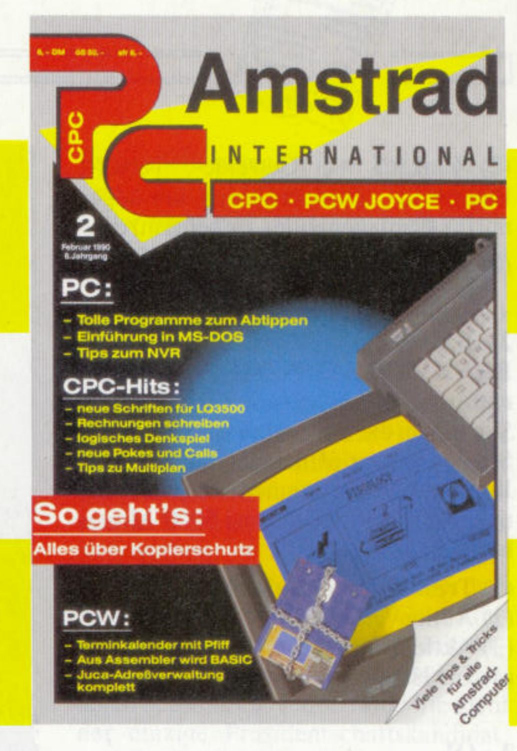
Laden Sie unter BASIC2 die Datei PRES.BAS, und starten Sie das Programm. Es können bis zu 6 Spieler am Rennen um den Präsidentensessel teilnehmen. Versucht sich nur einer an dieser Aufgabe, so sind die Sabotageakte natürlich nicht durchführbar.

Das Programm ist auf der DATABOX in der üblichen Form mit allen für das Spiel notwendigen Files enthalten. Doch nun viel Glück auf dem langen Weg zum Präsidenten der Vereinigten Staaten!

(Christian Stein/jf)

PCW
DATABOX





PC
DATABOX



DATABOX ist der preiswerte Software-Service Ihrer Zeitschrift
DATABOX enthält lauffähige Programme für Ihren Einsatz
DATABOX lohnt sich auf jeden Fall - Monat für Monat

PCW

Juca Teil 3
Ab jetzt sind Sie im Besitz der kompletten Joyce-User Club-Adreßverwaltung und können somit alle ihre Funktionen benutzen. Die bisher fehlenden Dateien können Sie dieser DATABOX entnehmen.

Kalender
Alle Jahre wieder wird ein neues Kalenderprogramm veröffentlicht, welches noch mehr Funktionen als der Vorgänger bietet. Trotzdem halten wir es nicht für nötig, Ihnen dieses Programm vorzuenthalten.

Von Assembler nach Basic...
...oder vielleicht besser: Wie erstelle ich in Basic einen Datalader? Mit unserem Programm Hexdat können Sie dies schnell und ohne viel Hin und Her praktizieren.

Analoguhr in Basic
Wenn Sie im Besitz der Basic-Erweiterung von Gerdes sind, wird Sie diese kleine Routine sicherlich interessieren. Sie ermöglicht es Ihnen, in eigenen Programmen eine Uhr **in beliebiger Größe** auf dem Bildschirm ausgeben zu lassen.

Als Bonusprogramm haben wir diesmal ein Spielprogramm mit dem Namen »Verbindungen« herausgesucht. Lassen Sie sich überraschen.

PC

MS-DOS – die ersten Berührungen
SUCH.BAT ist eine Batch-Datei, die alle Verzeichniseinträge Ihres Datenträgers nach beliebigen Zeichenfolgen durchsucht und diese in einer Datei ablegt.

J.R. in die Karten geschaut
"For President..." entführt Sie in das harte Leben eines Präsidentschaftsanwärters, der mit Geld und Macht das höchste Amt anstrebt. Eine Wirtschaftssimulation in Basic 2.

Strukturierte Programmierung
Mit UPCASE können alle nur denkbaren Programmcodes auf einfachste Weise in eine lesbare Form gebracht werden. Mit etwas Fantasie läßt sich der Einsatzbereich allerdings noch erweitern.

Gleich drei Programme haben wir Ihnen in diesem Monat in der Rubrik Bonusprogramm anzubieten. Sie sind für das Spiel in einsamen Stunden gedacht. Solitaire, Zahlenpuzzle und Black Jack werden Ihnen unterhaltsame Stunden bereiten.

Einzelbezugspreis für DATABOX:
PCW - 3-Zoll-Disk. PC - 5 1/4-Zoll-Disk.

24,– DM
Wenn Sie über den DMV-Verlag bestellen, gilt folgendes:

Inland:		**Ausland:**	
Einzelpreis	24,– DM	Einzelpreis	24,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**28,– DM**	**Endpreis**	**30,– DM**

Das DATABOX-Abo kostet:

Als Diskette für 1/2 Jahr (6 Lieferungen)	
Im Inland und West-Berlin	150,– DM
Im europäischen Ausland	160,– DM
Im außereuropäischen Ausland	180,– DM

Als Diskette für 1 Jahr (12 Lieferungen)	
Im Inland und West-Berlin	300,– DM
Im europäischen Ausland	320,– DM
Im außereuropäischen Ausland	360,– DM

Bitte benutzen Sie für Ihre Bestellung die Abo-Karte.

Zahlungshinweise:
Am einfachsten per Vorkasse (Verrechnungsscheck) oder als Nachnahme zuzüglich der Nachnahmegebühr. (Bei Lieferungen ins Ausland ist Nachnahme nicht möglich.)

Widerrufsrecht:
Jeder Abonnent hat das Recht, seine Bestellung innerhalb einer Woche beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich zu widerrufen. Die rechtzeitige Absendung des Widerrufschreibens genügt zur Fristwahrung.

Bitte benutzen Sie die Bestellkarte

DMV-Verlag · Postfach 250 · 3440 Eschwege



Berlin

Ihr Computer-Partner für



PC's/AT's·C 64/128
Amiga·PCW
Computer·Drucker
Zubehör·Software
Spiele·Service

W. Müller & J. Kramke GbR
Schöneberger Str.5 · 1000 Berlin 42
Tel. 030-752 91 50/60
Mo.-Fr. 10-18 Uhr. Sa. 10-13 Uhr



Ihre
COMPUTEREI

Schneider
COMPUTER DIVISION

Hardware
Software
Beratung
Literatur

Tempelhofer Damm 120
1000 Berlin 42
Am U-Bhf. Tempelhof
Tel. 7 52 20 91

Löhne/Ostwestfalen



Computer- & Softwarezentrum für Norddeutschland:
AMSTRAD-, SCHNEIDER- & VORTEX Regionalhändler & **SERVICE-CENTRALE.** Sämtl. Computer, Drucker, Peripherie & Zubehör v. A–Z, EDV Papier etc. • Discs
Fritz OBERMEIER COMPUTER•TELEFAX•BTX•HIFI•VIDEO•TV•
+ NEC•EPSON•TANDON•BROTHER•SEIKO•OKI•STAR•LOCO•etc.
am Bahnhof•Bünder Straße 20•4972 LÖHNE 1•Tel. 0 57 32 61 26/32 46

Nürnberg



Basel

AMSTRAD/SCHNEIDER
Vertragshändler

Computer Knüppel AG
Computer und Büromaschinen
Riehenring 81 (MUBA)
4058 Basel
Telefon (061) 691 1262
Fax (061) 691 0051

Castrop-Rauxel



Anzeigenschluß für die Ausgabe 5/90 von PC International ist der 7.3.90

Erscheinungstermin ist der 25.4.90

Eintragungen im Händlerverzeichnis, nach Städten geordnet, kosten je mm Höhe 6,– DM bei einer Spaltenbreite von 58 mm. Einträge möglich mindestens 6 × innerhalb eines Insertionsjahres.

Nähere Informationen:
DMV-Verlag
Wolfgang Brill
Telefon (0 56 51) 8 09-3 80

PC Amstrad INTERNATIONAL

ABONNEMENT

immer und überall

Ein Abonnement ist praktisch und gewährt zusätzlich noch Preisvorteile.

Das Abonnement PC Amstrad International ist preiswert zu haben:

Im Inland bzw. West-Berlin
6 Ausgaben = 33,– DM
12 Ausgaben = 66,– DM

Im europäischen Ausland
6 Ausgaben = 48,– DM
12 Ausgaben = 96,– DM

Im außereuropäischen Ausland
6 Ausgaben = 60,– DM
12 Ausgaben = 120,– DM

Bitte benutzen Sie die Bestellkarte.

Widerrufsrecht:
Jeder Abonnent hat das Recht, seine Bestellung innerhalb einer Woche beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich zu widerrufen. Die rechtzeitige Absendung des Widerrufschreibens genügt zur Fristwahrung.

DMV-Verlag · Postfach 250 · 3440 Eschwege



Biete Software

Astrologie mit Computer
International geschätzte Astrologenprogramme, professionelle Deutungsprogramme, Lernprogramme für Anfänger, Handschriftanalyse, Bio-Rhythmus, Astro-I-Ging.
Info gegen DM 2,– in Marken.
Astron, K.W. Bonert,Peter-Marqu.Str.4a 2000 Hamburg 60 G

CPC-PD SPIELE + ANWENDERPRG. LISTE 100 PF bei: Peter Breuker, Rektenstr. 10, 4930 Detmold 1

Wirtschaftliche Programme für die Arztpraxis auf dem Schneider CPC, Joyce, PC
Fa. EFFEKTA, Am Wiggert 9c 45 Osnabrück, 05 41/44 24 16 G

■■■■■■■■■■■■■■■■■■
Lohn- und Einkommensteuer 89:
STEUERMAT 89: Analyse aller Einkunftsarten, jährl. Aktu., Tabelle, Ausdruck aller Angaben auf Hauptantrag u. Anlagen N / KSO 98,– DM
Einnahme-Überschuß-Rechnung:
FIBUMAT: 142,– DM
FORMULARPRINT: 49,– DM
f. IBM-komp. PC/AT oder JOYCE
Demo: je 15,– DM, (3,5"- und 3"-Disk: 5,– DM Aufpreis)
EDV-SYSTEME-FARIN, FF-Software, Neuenhauserstr. 76, 4460 Nordhorn
TEL 0 59 21/3 94 22 FAX 0 59 21/3 36 22
■■■■■■■■■■■■■■■■■■ G

**** Dias ordnen mit Computer ****
CPC 464/664/6128, JOYCE und PC bis zu 100000 Dias; Suchzeit 1 Sekunde. Info gegen Rückporto bei: Dipl.-Ing. W. Grotkasten, Birnenweg 6, 7060 Schorndorf
Tel. 0 71 81/4 28 46 G

BONZO'S SUPER MEDDLER VS 4.0: DAS VIELSEITIGSTE PROGRAMM FÜR BAND-DISK-KOPIEN (alle CPCs). Auch für allerneuste Kopierschutztechniken! 3"-Disk mit über 1000 Kopierhinweisen nur DM 65,– + Versandk. Ausführl. Infos gg. Freiumschlag: MARTINA HIPPCHEN,PF 10 09 66, 5000 KÖLN 1 G

ENDLICH! PD-SOFTWARE AB DM 1,–
Für IBM/Kompat. Kat.-Disk. gratis.
M. Karbach - Remscheiderstr.18 5650 Solingen 1 G

ACHTUNG !! ERWACHSENE
10 Disketten 5 1/4 oder 3 1/2 mit delikaten Programmen f. IBM + komp. nur DM 35,– VScheck oder bar an M. K A R B A C H, Remscheiderstr.18, 5650 Solingen 1 G

#################
Software für JOYCE je 98 DM !
-FiBu (profess. Prog., DATEV-Konten)
-Finanzmathe: Zinsen, Annuitäten, Raten, Tilgungspläne (20 Prog.)
Kellmann, Wilhelmstr.71, 44 Münster
#################

WAHNSINN !!! Schneider-CPC-Original-Soft- und Hardware bereits ab 3 DM. Liste gratis bei Dieter Köhler, Brandstätterstraße 26, 8501 Cadolzburg

GT65, Rombox, Pascal MT+, CBasic, MBasic, Modula-2, Wordstar, Spiele Disks, Bücher, Zeitschriften.
Olaf Orth, Blücherstr. 2, 3500 Kassel, Tel. 05 61/5 44 73 Preise VS

################
PD-SERVICE-LAGE bietet Ihnen:
################
Public-Domain & Shareware z.B:
über **870 ! DEUTSCHE Programme !**
über **370 ! Disk PD-SPIELE** und
neuste INTERNAT. Programme !!!
Kopierkosten: 4,50--2,70 DM !!
Alleinvertrieb: GEDEY-CAD 4.03 VON WOLFGANG MAIER
Kat. f. IBM/kompat. g. 1,80 DM Porto
PD-SERVICE-LAGE: Bernd Schulz Postfach 17 43, 4937 Lage/Lippe
Auch alles auf 3,5" lieferbar !
NEU bei uns: Low-Cost-Software !
################ G

■ CPC & JOYCE PUBLIC DOMAIN ■
● Ein Info ist für 2x100Pf erhältlich ●
■ PDI, Pf 1118,D-6464 Linsengericht**■** G

Wegen Systemwechsel verkaufe ich meine umfangreiche Software für den CPC 6128. STAR-WRITER, FIBUSTAR-PLUS, DATEI-STAR, MICA usw. Daneben Schneider CPC-International ab 86, Sonderhefte mit Disk., viele Bücher und Programmbeschreibungen.
Tel. 0 70 22/5 20 06

Lohn- und Einkommensteuer 1989
Druckerausgabe + Datensicherung. Ausführliche Anleitung. Info 2 DM
3"-Disk für CPC 79,– DM + VP. Versand gegen Vorkasse oder NN. 89er-Aktualisierung 35,– DM. S. Teurich, Mesternstraße 6, 4952 Porta Westfalica 3 G

VERKAUFE FÜR SCHNEIDER 6128 MULTIPLAN, WORDSTAR 3.0, dBASE U. DATAMAT MIT 5 BÜCHERN (NEUP.1000 DM) ZUSA FÜR 600 DM, AUCH EINZELN ABZUGEBEN. CHIFFRE 89/00114 PC

Diskpara + MSCopy CP/M 3.0: 45,– Pro-Design + Erw.disc: 35,– Bonzo's Super Meddler 3.0: 30,– alles Originale mit Handbuch
Bücher : Peeks & Pokes : 10,– 6128 Intern: 25,–; Basic 6128: 15,–; Masch.sprache 6128 : 15,– Databox Artworx:8,– 02 21/44 26 33

Biete Hardware

ACHTUNG 50% Rabatt! CPC 6128 + GT 65 + DMP 2000 + Plotter + Scanner (Fisher-Technic) 1111,–
Tel: 02 11/68 16 08 erst ab 15 Uhr

Geprüfte Gebrauchtgeräte mit Garantie * Neue und gebr. Schneider und Amstrad PC*Floppy/Festplatten/ PC-MM und Farbmonitor* Neue und gebr. CPC/Floppy/Drucker* Ankauf bei Systemwechsel*Reparaturservice*Manfred Kobusch, Bergenkamp 8, 475 Unna, 0 23 03/1 33 45 G

Verk. Schneiderware (Centronics, Echtzeituhr, Basispl.) u. A/D-Wandler. Ab 18 Uhr Tel. 0 74 76/85 72

neuwertiger CPC464 (Farbe) + DD1 + GP500CPC + Papier + Kabel + 3 Handb. + Joy + Adapter + ca 40 Disks + 17 Cass. Preis VS
Tel.: 0 56 23/51 77

Lichtgriffel nur DM 49,–
Versand gegen Scheck/Nachnahme
Info gratis! Computer angeben!
Anschluß an jeden (!) Computer möglich. Standardversion für Amstrad lieferbar. Firma Klaus Schißlbauer, Postfach 1335R, 8458 Sulzbach, Tel.: 0 96 61/65 92 oder 09 41/99 99 15 bis 21 Uhr G

Verkaufe Amstrad 664 mit Farbmon. usw*Ruft an*0 65 43/38 71

Joyce PCW256 wg. Systemwechsel für 700 DM zu verkaufen. Original-Software für 50%, Liste anford.
F. Sterner, Th. Mann Str. 3, 6085 Nauheim, 0 61 52-6 35 75

CPC 6128 + ZUSATZLAUFWERK + DISKETTEN + SOFTWARE + LITERATUR PREIS DM 600, TEL 0 89/3 20 34 34

CPC464 + DDI + ZUBEHÖR 02 51/79 37 96

STARDRIVE 5 1/4"; Devpac-Assembl. Compak-Paket; 20 Disk m. Softw. kompl. 400 DM, 0 52 24/37 15

MIDI-Interface + Sequenzer-Software: 130,–; Scanner für alle Drucker + Treibersoftware 90,–; TES Lightpen + Grafiksoftw. 25,–; Comm.mouse 20,– 02 21/44 26 33

Schneider PC 1640 HD. 1 Laufwerk, 20 MB Festplatte, Hercules-kompatibel, Monochrom-Monitor, Tel. 0 91 90/12 94

Zu verkaufen: CPC464/Farbe + DDI + F-1X + SP512 1300 DM. EINZELVERK. TEL.: 0 22 27/34 88 ab 15h.

Suche Software

DFÜ-TERMINAL-PROG. f. Joyce Chiffre 90/00116PC

S. "FUGGER" F. 464, 0 63 55/20 31

Suche Hardware

3"-FLOPPY F. CPC464 (ERSTLAUFW) U. SPEICHERERW. GESUCHT 0 82 34/21 12

Suche deutsches Handbuch für Multiplan 1.06 der Microsoft Corporation, 1981, 1983
Chr. Mowka, Grandstr.13, Essen 11
Tel.: 02 01/69 28 46
Wer kann helfen?

Verschiedenes

PC 1512/1640 - Anwender ! Werden auch Sie Mitglied in unserem USER-CLUB. Info gg. Rück porto von R.Knorre, Postfach 20 01 02, 5600 Wuppertal 2

upLink bringt Daten u. Progr. vom CPC u. JOYCE (m. CPS 8256) direkt zum PC/XT/AT. Transfer über die parallelen Schnittstellen.
Komplett 198 DM - NN.
Info. / Bestellung: CONCEPTION GmbH, Hubertusweg 14, 2000 Hamburg 61, Telefon: 0 40/58 45 03 Telefax 0 40/58 73 52 G

SUCHE LAUFW.FÜR 6128 + TAUSCHP. F. CPC-DISC,FÜRHOFF 0 40-2 81 29 46

SUCHE JOYCE-ANWENDER IN BREMEN UND UMZU! BIN ANFÄNGER. R.LAPPE H.D.GRAMBKER KIRCHE 20 2820 BREMEN 77 TEL:64 04 48(18.00)

2000 Computerbücher, Gratis-katalog, 80 S., T. 05 71/5 21 41 G

PC ZEITSCHR.V.8/85-12/89 + DATAB 7/86-5/89 + 8-9/85 NUR GESAMTABG = 400 DM + PORTO SONDERH.1-7 + DATA KAS + DISK NUR GESAMTABG. = 200 DM + PORTO, TEL.0 43 42/49 43, 17-18 UHR -AB 18 UHR 0 43 42/35 09

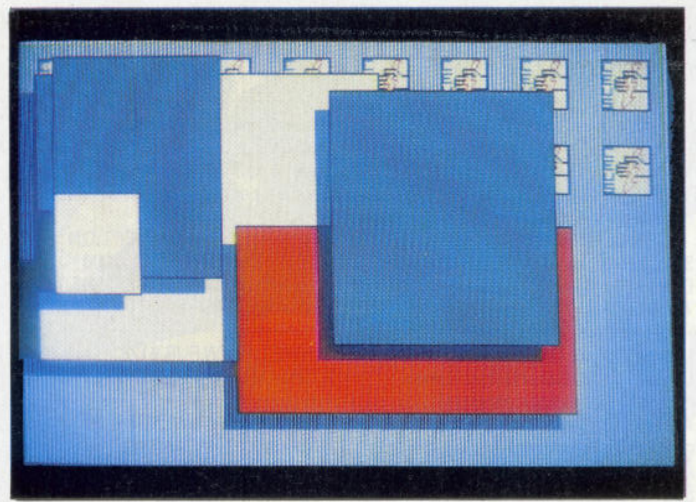
Tausch

Achtung CPC-464 3" bzw.3 1/2" oder Kassette: Suche Country Cottages sowie weitere gute Software (besonders Strategie-Programme wie Kaiser usw.) Liste an: Thomas Brodersen, Saturnstr. 5, 5650 Solingen 11

Club

USER-CLUB für JOYCE-Anwender! Info gegen Rückporto von: JOYCE-USER-CLUB c/o Jc. Berghof, Roseggerstr.5, 5600 Wuppertal 2

*** MISTERSOFT COMPUTER CLUB ***
»Der« Software-User-Club für aktive PC-Anwender! Riesiges Angebot an guter Public Domain und Shareware, auch für EGA!!!
KEIN MITGLIEDSBEITRAG!! Infos: Ralf Jodl, Hessenstr.15, D-8912 Kaufering. (Bitte Rückporto!)



Die Götter werden verrückt, wenn sie sehen, was CEUS auf dem CPC veranstaltet: eine komplette grafisch unterstützte Benutzeroberfläche

»PC International« 3/90 erhalten Sie ab:



PCW:

PCW als Speichergigant. Nicht etwa eine Speichererweiterung, sondern vielmehr ein Vorschlag, wie Sie mit zwei 720-kByte-Laufwerken an Ihrem PCW arbeiten können, erwartet Sie im nächsten Heft.

Für LocoScript 1 ein alter Hut, aber für die Zweierversion noch nicht veröffentlicht. Endlich ist es jetzt auch mit LocoScript 2 möglich, mit 43-Spur-Disketten im Laufwerk A zu arbeiten.

Benötigen Sie ein Datei-, Text- und Tabellenkalkulationsprogramm für Ihren PCW? Mini Office Professional in der Plus-Version könnte eine Hilfe für Sie sein. Einen ausführlichen Test können Sie der nächsten Ausgabe der PC-Amstrad entnehmen.

CPC:

Das Problemkind Nummer 1 des Computers ist der Drucker, das Problemkind Nummer 1 des Druckers ist die Hardcopy. Wenn tatsächlich eine von einem Programm versprochene Grafik auf dem Papier erscheint, ist es, als fiele Neujahr auf Silvester. Wir geben Ihnen die notwendigen Grundlageninformationen, damit Sie Ihr persönliches Hardcopyprogramm anfertigen können; denn: Selbst ist der CPC-Besitzer.

Sie blicken neidisch auf die PC-Besitzer, die spielend leicht mit komfortablen Benutzeroberflächen wie Windows umzugehen wissen? Lassen Sie sich überraschen, wozu auch Ihr CPC in der Lage ist. CEUS ist das Zauberwort. Begleiten Sie uns in unserer neuen Serie in die Geheimnisse der grafisch orientierten Benutzeroberflächen. Auf das die PC-Besitzer vor Neid feuchte Augen bekommen...

Zu zweit spielt es sich am besten. Unter diesem Motto wurde das Programm UNITER erstellt, eine strategische Auseinandersetzung für kühle Denker.



Taktisch geht es zu bei Uniter, einer heißen Auseinandersetzung, die die Gehirne in höchste Aktivität versetzen wird

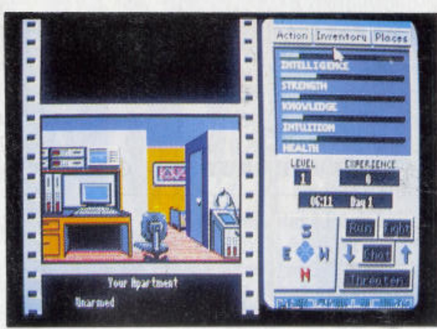
PVS – Programmierte Vektor-Schrift. Wer Schriften per Pixelzoom vergrößert, der bekommt stets eine zwar große aber auch dementsprechend "blockige" Schrift zu sehen. Berechnete Vergrößerungen dagegen sind variabler und schöner anzusehen. Wie es geht? In Ausgabe 3/90 der PC Amstrad können Sie es nachlesen.

PC:

Im dritten Teil unserer MS-DOS-Einsteiger-Reihe befassen wir uns mit der Konfigurationsdatei CONFIG.SYS und deren speziellen Möglichkeiten zur Voreinstellung Ihres Computersystems.

Ins Detail geht es dann, wenn wir das NVR zum zweitenmal durchleuchten. Lesen Sie in unserer nächsten Ausgabe, wie man diesen Speicher manipuliert.

Daß das Softwarehaus Accolade mittlerweile eine bunte Palette an verschiedenen Unterhaltungsprogrammen auf den Markt gebracht hat, war uns Anlaß genug, die interessantesten Spiele einem Test zu unterziehen.



The Third Courier, ein Agententhriller – nur eines von einer ganzen Reihe von Neuerscheinungen aus dem Hause Accolade

Die Kleinstbenutzeroberfläche RUN gibt Ihnen auf komfortable Weise die Möglichkeit, alle nur lauffähigen Programme zu starten oder einfach nur zu listen.

Das Wort GEM-Installation veranlaßt so manchen Computerbesitzer die Flucht zu ergreifen. Die vom Handbuch arg vernachlässigten Tips und Finessen, wie man diese Benutzeroberfläche komfortabel und vor allem variabel in den Griff bekommt, wollen wir nachholen.

Stichwort Hardcopies: Wir haben für Sie das Programm Pizzaz unter die Lupe genommen und geben Ihnen als Zugabe ein komplettes Hardcopy-Programm auf die Databox.

DIE INSERENTEN

CSV Riegert ... 41
DMV ... 2,11,13,27,45,57,64,65,71,83,98,100,103,104
Dobbertin Elektronik ... 19
G + L electronic ... 79
Kosmalla + Partner ... 25
Kotulla ... 19
Krebs electronic ... 91
Rätz Verlag ... 79
Schuster Electronic ... 53
Stähle ... 15
Strauß Elektronik ... 37
Weber ... 15
van der Zalm ... 91


DMV
Daten- und Medienverlag